

LINUX JOURNAL

Since 1994: The Original Magazine of the Linux Community

SPONSORED BY



JUNE 2013 | ISSUE 230 | www.linuxjournal.com

WORKING WITH

ANDROID

USE
MultiROM
to Multi-Boot
Your Nexus 7

CODE
ON THE
RUN
with AIDE
for Android

TEST
Your
Android
Implementation



A LOOK
AT WORMS
AND LINUX

BUILD A
REDUNDANT
WEB CLUSTER
WITH RPi

A GUIDE TO
UNDERSTANDING
UNICODE

OPINION:
21ST CENTURY
DEVOPS—AN
END TO
STATIC BUILD
AND DEPLOY SCRIPTS

How To: Get Started Mining Cryptocurrency



Put Your Android App Under the Microscope with New Relic

See what you've been missing.

A central illustration of a microscope with an Android robot head integrated into the eyepiece. Four teal circular callouts are connected to different parts of the microscope by thin lines. The callouts contain text describing the benefits of New Relic for Android apps.

GET TOTAL VISIBILITY
INTO YOUR ENTIRE
APPLICATION

IMPROVE
PERFORMANCE
& SPEED

ADJUST YOUR
MONITORING
SETTINGS AND GET
AS GRANULAR
AS YOU NEED

SPOT
CODE-LEVEL
ERRORS

newrelic.com/android

zStax

ZFS Unified Storage

Father and son take their need for speed from the track to the data center.

Is your current storage solution slowing down your Tier 1 applications?

Steve and Tommy Scherer, two Silicon Mechanics experts, know that storage performance is crucial for today's datacenter deployments. The zStax StorCore 104 unified storage appliance exceeds the performance requirements of today's business without restricting customers to legacy proprietary hardware.

The zStax StorCore 104 is specifically tailored to meet the storage requirements for datacenter technologies like virtualization, cloud computing, VDI, and software-defined storage architectures at a fraction of the cost of legacy storage vendors.

Take a ride on the zStax StorCore 104. Best-in-class storage, full of win.



Steve Scherer

Tommy Scherer

ANDROID

FEATURES

72 **AIDE—Developing for Android on Android**

Code on the run, no matter where you are, with AIDE.

Joey Bernard

88 **Compatibility Test Suite—Test Your Android Implementation**

Test your Android implementation before you go to customers.

Nitish Tiwari

98 **Multi-Booting the Nexus 7 Tablet**

Stop flashing your device over and over just because you want to try something new. MultiROM to the rescue!

Bill Childers



ON THE COVER

- Use MultiROM to Multi-Boot Your Nexus 7, p. 98
- Code on the Run with AIDE for Android, p. 72
- Test Your Android Implementation, p. 88
- A Look at Worms and Linux, p. 106
- Build a Redundant Web Cluser with RPi, p. 52
- A Guide to Understanding Unicode, p. 40
- Opinion: 21st Century DevOps—an End to Static Build and Deploy Scripts, p. 116
- How To: Get Started Mining Cryptocurrency, p. 58

INDEPTH

106 Worms and Linux

A look at how worms work, and some Linux-specific worms.

Himanshu Arora

OPINION

116 21st-Century DevOps—an End to the 20th-Century Practice of Writing Static Build and Deploy Scripts

Embracing 21st-century DevOps means letting go of 20th-century practices.

Tracy Ragan

COLUMNS

40 Reuven M. Lerner's At the Forge

Unicode

48 Dave Taylor's Work the Shell

Cribbage: Pairs and Three of a Kinds

52 Kyle Rankin's Hack and /

Two Pi R 2: Web Servers

58 Shawn Powers' The Open-Source Classroom

Prospecting for Ones and Zeros

122 Doc Searls' EOF

Android's Limits

IN EVERY ISSUE

8 From the Editor

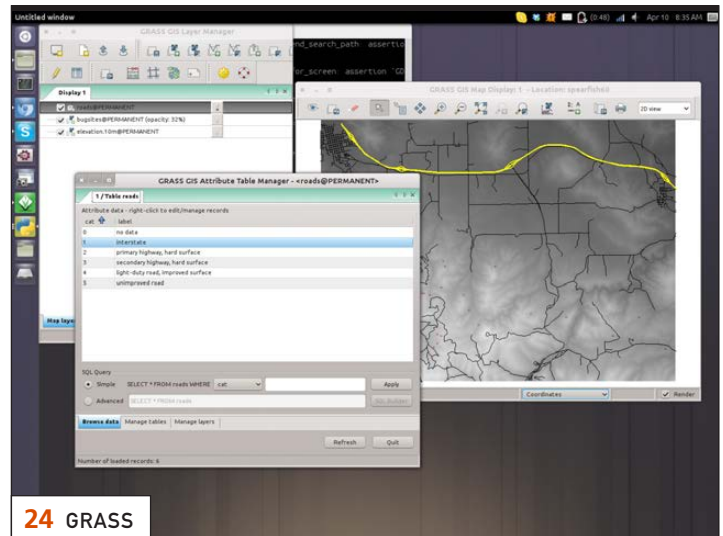
12 Letters

18 UPFRONT

38 Editors' Choice

68 New Products

127 Advertisers Index



LINUX JOURNAL™

Subscribe to
Linux Journal
Digital Edition
for only
\$2.45 an issue.



ENJOY:

Timely delivery

Off-line reading

Easy navigation

Phrase search
and highlighting

Ability to save, clip
and share articles

Embedded videos

Android & iOS apps,
desktop and
e-Reader versions

SUBSCRIBE TODAY!

LINUX JOURNAL

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

Contributing Editors

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

Publisher Carlie Fairchild
publisher@linuxjournal.com

Director of Sales John Grogan
john@linuxjournal.com

Associate Publisher Mark Irgang
mark@linuxjournal.com

Webmistress Katherine Druckman
webmistress@linuxjournal.com

Accountant Candy Beauchamp
acct@linuxjournal.com

**Linux Journal is published by, and is a registered trade name of,
Belltown Media, Inc.**

PO Box 980985, Houston, TX 77098 USA

Editorial Advisory Panel

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

Advertising

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

Subscriptions

E-MAIL: subs@linuxjournal.com
URL: www.linuxjournal.com/subscribe
MAIL: PO Box 980985, Houston, TX 77098 USA

LINUX is a registered trademark of Linus Torvalds.

High Performance, High Density Servers for Data Center, Virtualization, & HPC

On-board 10 Gigabit Ethernet and Infiniband for greater throughput in less rack space

The Intel® Xeon® Processor E5-2600 family powers the highest-density servers iXsystems has to offer. The iXR-1204 +10G features dual onboard 10GigE + dual onboard 1GigE network controllers, up to 768GB of RAM and dual Intel® Xeon® E5-2600 family processors, freeing up critical expansion card space for application-specific hardware. The uncompromised performance and flexibility of the iXR-1204 +10G makes it suitable for clustering, high-traffic webservers, virtualization, and cloud computing applications - anywhere you need the most resources available.

For even greater performance density, the iXR-22X4IB squeezes four server nodes into two units of rack space, each with dual Intel® Xeon® E5-2600 Family Processors, up to 256GB of RAM, and an on-board Mellanox® ConnectX QDR 40Gbp/s Infiniband w/QSFP Connector. The iXR-22X4IB is perfect for high-powered computing, virtualization, or business intelligence applications that require the computing power of the Intel® Xeon® Processor E5-2600 family and the high throughput of Infiniband.

iXR-1204 +10G

- Dual Intel® Xeon® Processors E5-2600 Family
- Intel® X540 Dual-Port 10 Gigabit Ethernet Controllers
- Up to 16 Cores and 32 process threads
- Up to 768GB Main Memory
- 700W Redundant high-efficiency power supply

iXR-22X4IB

- Dual Intel® Xeon® Processors E5-2600 Family per node
- Mellanox® ConnectX QDR 40Gbp/s Infiniband w/QSFP Connector per node
- Four server nodes in 2U of rack space
- Up to 256GB Main Memory per server node
- Shared 1620W Redundant high-efficiency Platinum level (91%+) power supply



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

E5-2600

HIGH &
Throughput
INCREDIBLE
Performance Density

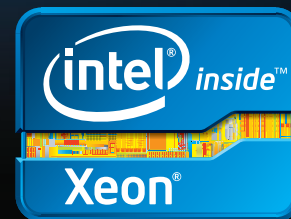


iXR-1204+10G: 10GbE On-Board

4 Server
Nodes
in 2U



iXR-22X4IB



Call iXsystems toll free or visit our website today! **1-855-GREP-4-IX** | www.iXsystems.com



SHAWN POWERS

The Face of a Toaster

Science-fiction geeks have very specific definitions for their (okay, “our”) jargon. A cyborg is a hybrid mechanical and organic creature. An AI (artificial intelligence) is a purely electronic device or program capable of thought. And an android is a purely mechanical and electronic device with a human appearance. In *Star Trek* terms, the Borg are cyborgs, and Data is an android. Then Google came and messed everything up. “Android” is now synonymous with the operating system on phones and tablets everywhere. Unless it’s an obscure social commentary on square shapes and human obesity, I think it’s safe to assume Google just went with a cool-sounding, futuristic name. This month, we focus on Google’s Android, the Linux-based operating system that’s taking over the planet.

Reuven M. Lerner starts out the issue with the universal translator

of the programming world—namely, Unicode. Although it makes life easier all across the planet, working with Unicode can be challenging. Reuven addresses some of those challenges, and shows how to work through them. Dave Taylor helps us work through some things too as he continues his series on the *Cribbage* script we’ve been working on. Whether you want to learn scripting, or better understand *Cribbage*, it’s an awesome series.

Kyle Rankin brings us another Raspberry Pi article this month, this time doing fault tolerance with redundant Web servers. Kyle’s articles always make me hungry for knowledge, but the past couple months, his titles have made me hungry in general—Raspberry Pi, indeed. I, on the other hand, talk about cryptocurrency this month. It’s been a while since I’ve mentioned Bitcoins or Litecoins, and with the recent bubble, perhaps it’s time

to dust off those miners and start generating cash. It's not a surefire way to get rich, but you might be able to earn enough income to buy some Raspberry Pi devices and keep up with Kyle!

Generally, developers develop computer programs on computers. They also develop mobile applications on computers. Joey Bernard shows us AIDE this month, which is a nifty way to develop Android apps while using the Android OS! If you've ever wanted to do some coding while on the road, but you're stuck with only your tablet, you'll want to check out this article. Nitish Tiwari delves into development this month as well, but he discusses testing side of things. If you want to port your application to new hardware, go directly to his article and start reading.

Bill Childers and I are kindred spirits when it comes to mobile technology. I think his pile of goodies is a little taller than mine, but thankfully, he shares his experiences with us all. A couple months back, I wrote about my Nexus 7 tablet. Bill goes in a slightly different direction with his Nexus 7, and he explains how to boot in to completely different ROMs! Whether you want to try

CyanogenMod or Ubuntu on your tablet, MultiROM is your tool, and Bill describes how to use it.

At times, we can get arrogant about security as Linux users. Himanshu Arora looks deeper into viruses and worms. Are we safer by design? Are we setting ourselves up for disaster? Himanshu's article is a must-read for anyone interested in the debate. And finally, Tracy Ragan's opinion piece on DevOps is one that hits home to an old-timer like me. The scale of data centers is such that the traditional static scripts and hacks are being phased out by newer, truly automated systems.

If you think Android is a little green robot that lives in your phone, or if you think Google has blasphemed all of science fiction, this Android issue of *Linux Journal* should be enjoyable. I can't bring myself to make a cheesy comment like, "set phasers to awesome", so I'll just hope you enjoy this issue. We think it's awesome, phasers or no. ■

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



Dedicated server KS1

\$39.00/month

free setup

Processor

Intel Core i3 2130 (SANDY BRIDGE)
2 Cores (4 THREADS)
3.4 GHz+

RAM

8 GB DDR3

Hard Drive

2 x 1 TB SATA2
RAID SOFT (0/1)

Guaranteed Bandwidth

100 Mbps
5TB traffic/month

Dedicated server SP 2

\$89.00/month

free setup

Processor

Intel E3 - 1245v2 (IVY BRIDGE)
4 Cores (8 THREADS)
3.4 GHz+ (3.8 GHz TURBO BOOST)

RAM

32 GB DDR3

Hard Drive

2x 2TB SATA3
RAID SOFT (0/1)

Guaranteed Bandwidth

100 Mbps Bandwidth included
Unlimited traffic

Dedicated server EG 3

\$179.00/month

free setup

Processor

Intel E5 - 1620 (SANDY BRIDGE)
4 Cores (8 THREADS)
3.6 GHz+ (3.8 GHz TURBO BOOST)

RAM

32 GB DDR3 ECC

Hard Drive

2x 3TB SATA3 + 80 GB SSD
MegaRAID 9271 6Gbps - 1 GB Cache -
CacheVault - CacheCade

Guaranteed Bandwidth

200 Mbps up to 1 Gbps
Unlimited traffic

Dedicated server mHG1

\$259.00/month

free setup

Processor

2x Intel Xeon E5606
2x 4 Cores
2x 2.13GHz

RAM

128 GB DDR3 ECC

Hard Drive

2x 600GB SAS 15k rpm
MegaRAID 6Gbps - Battery Kit

Guaranteed Bandwidth

300 Mbps up to 3 Gbps
Unlimited traffic

Dedicated Infrastructure for your Business

Unlimited Traffic &
Guaranteed Bandwidth



24/7 North-American
Support



Total Control



Private Cloud



Dedicated Servers



Dedicated Cloud

For more details:

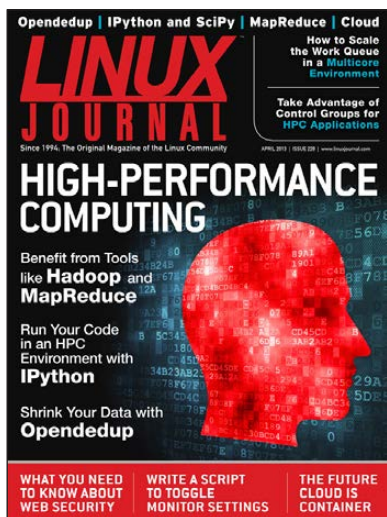


OVH.COM

Dedicated Infrastructures For Your Business

or contact us: **1-855-684-5463** (toll free)

letters



Linux and Windows File Servers

I've been looking through the April 2013 issue of *LJ's* Letters section with particular interest in the

uptake of Linux in schools, and I'd agree with most of what Lucian Macu and you have to say.

It's not just schools though where the adoption of Linux is perceived to be difficult. In my previous job with the Birmingham City Council, I was one of a few users who ran Linux on the desktop. I was on the UNIX team, so it seemed the most natural thing to do. However, initially the back-end home drives and so on were on Novell, and then they moved sadly to Windows. We ran Lotus Notes for e-mail, and I can say the native client for v8 worked very well with Fedora (can't remember the release).

In my current job with IGT, until recently, I have been forced to use the

truly dreadful Windows 7, although recently due to Windows 7 crashing and not starting up yet again (four times in two and a half years), I'm back with a Linux desktop in a totally Windows environment.

To try to streamline living in this environment, I started writing some login scripts that would cope with Novell (ncpfs), Windows (Samba CIFS and so on) and Linux (sshfs) back ends and create a mapped drive similar to a Windows desktop. It wouldn't be too hard to cope with a Mac desktop to these back ends too. Some embryonic code exists to detect OS X, but as I don't have a Mac currently, this is on hold.

If there is a mix of desktop systems, it is desirable to have a degree of uniformity to them. Personally, I find the Windows desktop clunky to use, but if I were a user familiar with how Windows desktops work, the idea of mapped drives rather than mounted filesystems grafted on at a mountpoint would be easier to adjust to.

Also, if I am a Linux laptop user, I would not want a remote home

directory mounting over my /home/ user, and if there are shared team filesystems, there still needs to be somewhere to put them, so on balance, I think the idea of using a drive letter isn't too bad for a desktop system. Of course, the drive letter is really just a mountpoint, but let's not tell the users that!

So far, development has halted a bit for various time-related reasons, but the work as it exists is freely available on my Web site:

<http://www.rainsbrook.co.uk/wiki/doku.php?id=rb:linuxloginscripts>.

One main block to progress has been the move by Linux from SMB to CIFS. I used to be able to use smbmount, but CIFS doesn't want to allow a user to run this without anything in fstab.

One of the golden rules I have tried to set is that all this drive mapping and setting mountpoints should be able to be run as the local user. I'd really like to avoid any root level work. Ultimately, if the login script lives in the user's local home directory, the cfg part could come from the netlogon shared directory for Windows or an HTTP link. If the cfg file is changed by



an administrator, there should be *no* requirement for the system admin to modify anything locally.

For many years now, I have been a passionate Linux user. I am constantly frustrated by the lack of progress on adoption in education and business. It's more than capable for use in either environment with some small developments.

Ultra Small Panel PC

- ARM9 400Mhz Fanless Processor
- Up to 1 GB Flash & 256 MB RAM
- 4.3" WQVGA 480 x 272 TFT LCD
- Analog Resistive Touchscreen
- 10/100 Base-T Ethernet
- 3 RS232 & 1 RS232/422/485 Port
- 1 USB 2.0 (High Speed) Host port
- 1 USB 2.0 (High Speed) OTG port
- 2 Micro SD Flash Card Sockets
- SPI & I2C, 4 ADC, Audio Beeper
- Battery Backed Real Time Clock
- Operating Voltage: 5V DC or 8 to 35V DC
- Optional Power Over Ethernet (POE)
- Optional Audio with Line-in/out
- Pricing starts at \$375 for Qty 1

PPC-E4+

2.6 KERNEL

The PPC-E4+ is an ultra compact Panel PC that comes ready to run with the Operating System fully configured on the onboard flash. The dimensions of the PPC-E4+ are 4.8" by 3.0", about the same as that of popular touch cell phones. The PPC-E4+ is small enough to fit in a 2U rack enclosure. Apply power and watch either the Linux X Windows or the Windows CE User Interface appear on a vivid 4.3" color LCD. Interact with the PPC-E4+ using the responsive integrated touch-screen. Everything works out of the box, allowing you to concentrate on your application rather than building and configuring device drivers. Just Write-It and Run-It.

www.emacinc.com/panel_pc/ppc_e4+.htm

Since 1985
 OVER
28
 YEARS OF
 SINGLE BOARD
 SOLUTIONS

EMAC, inc.

EQUIPMENT MONITOR AND CONTROL

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

I'd hoped that with the Linux adoption by Novell, we might have seen more business/user developments in distros rather than just technical and architectural ones, as it is, Linux is more than capable, but it needs some bits bolted on to help it play well in an alien environment.

I'm not really sure where to go with this. I'd hoped that someone in the local LUGs would have picked up on this and offered to join in, but not much interest has been shown. I think I need to try to open this up and get more ideas in to solve some of the blocks that exist and maybe to clean up the code and so on. So if you know anyone, please point them to the Web page!

—**Andrew Stringer**

Well, it's posted as a Letter to the Editor, so all our readers will see your link. Perhaps it will go somewhere.

I will note that I was very excited when Novell turned to Linux as well. Unfortunately, like you, I haven't seen anything really come from it. I deal with Windows a lot in my day job, so in that regard, I feel your pain too. I try to look

at the heterogeneous computing environment as a learning opportunity, but it's still frustrating at times. I'm not able to use Linux as my main computer during the workday, but thankfully, I can have a virtual machine running full screen. Thanks again for your letter—Shawn Powers

Space Program on Twitter

The best thing I've done to re-ignite my enthusiasm for the space program is to join Twitter, so I could follow @Cmdr_Hadfield. He's the Canadian commander of the ISS for the next two weeks still, and he tweets down some amazing photos of places on Earth and the aurora and such. He's a big hero up here in Canada, of course.

Also, I hope you get your ISP problems sorted out soon, as I neglected to download the April PDF and epub files; I don't remember reading it either. I must have been out to lunch on April 1st.

—**Neal Murphy**

I love Commander Hadfield! I think his foray into social media has done more to excite the younger generation than years of marketing ever did. Oh, and

my ISP problems? Yeah, it seems to be getting better[CONNECTION LOST].

—Shawn Powers

The Open-Source Classroom Column

I've been a subscriber to *Linux Journal* for the past 5+ years, and I thoroughly enjoy working in Linux (Ubuntu/Debian). I believe that not enough is done to expose the children of South Africa to Linux from an early age. They are mostly educated on Windows/Office-based systems.

For a while now, I have been seriously considering starting a physical training center to teach them about Linux, open-source software, Raspberry Pi, Arduino and so on, but the details haven't been finalized yet.

The reason for this e-mail is to find out if there will be any issues if I call the center "The Open-Source Classroom". I know there probably are a million other choices, but the name is so appropriate for what I have in mind.

—Roelof Oelofsen

I responded right away when I got this, but I'll reiterate for everyone—yes, by all means, use the name! We chose the title for my column because it sounded cool, but it's not

something unique or specific enough to the magazine to worry about it.

Thank you for asking, but even more, thank you for your passion in regards to educating kids!—Shawn Powers

Linux Poetry

I write original pieces that I e-mail to friends and colleagues every Friday. This one, from this week, seemed like something you might like to "print" in *Linux Journal*. (I am a longtime subscriber and still have issue 1 wrapped in plastic!)

My title is "Linux Operations, Relations And X" to understand my position, go read the Specs.

They call me the LORAX, I speak about trees.
Rooted, Directed, Acyclic and Bs.

I speak about trees, for the trees have no tongues,
And I'm asking you, sir, at the top of my lungs -

I love backyard flowers, and gardens and seeds,
but Version Control Systems, use the trees in my needs.

It lets people find the files that go missing,
it lets me bring back bad mods, that others were dissing.
There's CVS, RCS, Subversion and others,
with features and options and ideology that smothers.

I search for myself, I search for my team,
for without simple GUIs,

[LETTERS]

They'll simply run out of steam.

I'm looking at Bazaar and I'm looking at Git,
I'm looking, and looking and looking more, a bit.

They call me the LORAX, I speak about trees.
I care about branches, and trunks and tags for these.
It must be easy and it must be simple,
it must make people smile and show me their dimple.

I already got Git, Git I already gat,
don't correct my conjugation, I don't wear that hat.
Bazaar they say is simpler,
they say that's what to get,
But I rely on my gut, my gut says get Git.

Git is bazaar, and Bazaar, I can get,
but for the non-data-arborist, both are a threat.

Git is from Linus, so it must be well done,
Linus did Linux, then Git just for fun.
I can insist, and I will, on my best rule of thumb
smart things from smart people really make me feel dumb.

When I smell Git's docs, I feel like a sneeze,
How will I ever find the forest, while looking at trees?

Still, they call me the LORAX, I speak about trees.
Rooted, Directed, Acyclic and Bs.

—Dennis Director

*Poetry for geeks is cool,
And Dr. Suess rocks,
So truly everyone wins!*

—Shawn Powers

Evernote

Just a follow-on from the Everpad article [see Shawn Powers' Everpad article in the April 2013 Upfront section]—I find that the Windows .exe version works just fine for me in Linux using Wine (1.5). But just as good, if not better, is the Evernote extension for Chromium (or Chrome) as well as the Evernote Web Clipper add-on for Firefox.

—Bill

I find myself using the Evernote Web app in Chromium more often than not. I'll admit, I haven't tried the Wine solution yet; I'm not sure why it never occurred to me! I still wish they provided a native Linux client, but at least the Web application is robust. Thanks for the tips!—Shawn Powers

Photo of the Month

When cleaning out a closet with old papers, I found the attached. How many memories this card has brought me. I think it would be a

At Your Service

SUBSCRIPTIONS: *Linux Journal* is available in a variety of digital formats, including PDF, .epub, .mobi and an on-line digital edition, as well as apps for iOS and Android devices. Renewing your subscription, changing your e-mail address for issue delivery, paying your invoice, viewing your account details or other subscription inquiries can be done instantly on-line: <http://www.linuxjournal.com/subs>. E-mail us at subs@linuxjournal.com or reach us via postal mail at *Linux Journal*, PO Box 980985, Houston, TX 77098 USA. Please remember to include your complete name and address when contacting us.

ACCESSING THE DIGITAL ARCHIVE: Your monthly download notifications will have links to the various formats and to the digital archive. To access the digital archive at any time, log in at <http://www.linuxjournal.com/digital>.

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

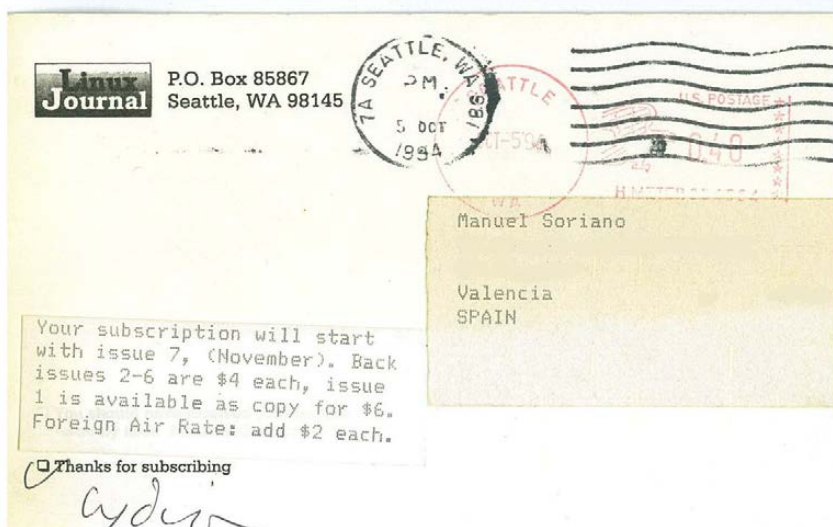
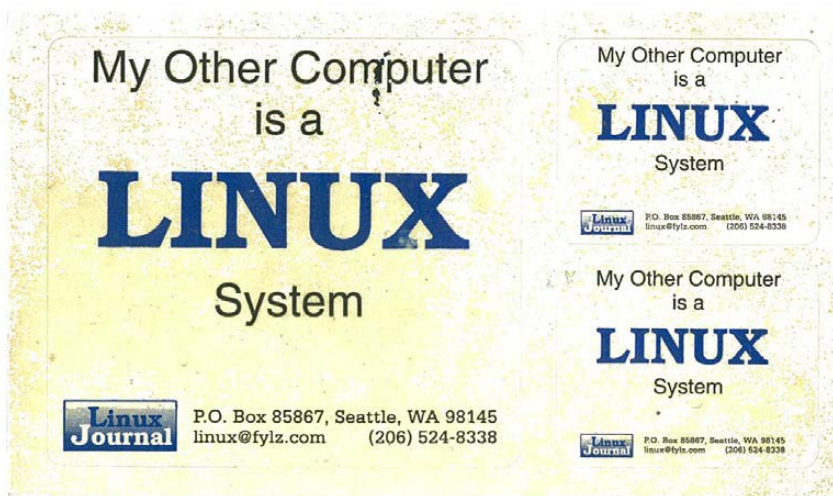
WRITING FOR US: We always are looking for contributed articles, tutorials and real-world stories for the magazine. An author's guide, a list of topics and due dates can be found on-line: <http://www.linuxjournal.com/author>.

FREE e-NEWSLETTERS: *Linux Journal* editors publish newsletters on both a weekly and monthly basis. Receive late-breaking news, technical tips and tricks, an inside look at upcoming issues and links to in-depth stories featured on <http://www.linuxjournal.com>. Subscribe for free today: <http://www.linuxjournal.com/newsletters>.

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: <http://www.linuxjournal.com/advertising>. Contact us directly for further information: ads@linuxjournal.com or +1 713-344-1956 ext. 2.

good idea to republish this card and send it to those who renew their subscriptions.

—Manuel Soriano-Vic Pérez



WRITE LJ A LETTER We love hearing from our readers. Please send us your comments and feedback via <http://www.linuxjournal.com/contact>.

PHOTO OF THE MONTH

Remember, send your Linux-related photos to ljeditor@linuxjournal.com!

diff -u

WHAT'S NEW IN KERNEL DEVELOPMENT

Recently, **Rob Landley** tried to remove a **Perl** dependency that has cropped up in the Linux build system...and tried...and tried. But, no one appeared to want his patches. Even **Andrew Morton** argued against removing the Perl dependency, in spite of the fact that Rob's patches replaced the code with smaller, simpler shell scripts.

It's unclear what the reluctance stems from—my understanding always has been that any chance to remove a dependency from the Linux build system was a chance worth taking. But apparently, the top kernel developers see a value in this particular dependency.

The e-mail thread was a bit of an eye-opener for me personally, because I'd started to consider Perl to be a legacy language, with no forward progress on Perl 6, and the language itself essentially stalled.

Microsoft's "secure boot mode" seems to be pitting kernel developer against kernel developer successfully. **David Howells** recently posted some patches to allow the kernel to

load Microsoft-signed cryptographic keys dynamically, and there was some support for the patch. But, **Linus Torvalds** wouldn't allow it, and there was some support for his rejection of it.

Linus' approach seems to be that he'll support anything that can provide genuine security to the user. And if some new technology is not designed to provide that security, he'll support it to the extent that it can be torqued into a position that provides genuine security. But, he won't support technologies that claim to add security when all they really do is take control of the system away from the user.

Evidently he feels that, as designed, "secure boot mode" does more to take control away from the user, than it actually does to secure the system. So he's rejected David's patches, as well as apparently any notion of catering to Microsoft as the sole key-signing authority for "secure boot mode" security keys.

According to **Thomas Gleixner** and others, the Linux kernel's

hotplug code had been developing races, undocumented behaviors and other problems. He posted a patch to rip the guts out of the beast and replace it with something that, to the user, would appear as a simple state machine.

Linus Torvalds liked the idea, but he wanted to be sure that the guts really were fully ripped out and replaced. He didn't just want to hide the problems. But as **Rusty Russell** pointed out, Thomas' patch represented only the first step, in which the horror was hidden from the user. The second step would involve cleaning up the internals themselves.

Rob Landley recently chafed at the increased security on the **kernel.org** server. Ever since the 2011 security breach, the kernel.org admins have implemented tighter access controls that limit the ways in which developers can contribute their work. Rob in particular missed the ability to rsync his code updates to a kernel.org account. As a result, he hadn't been able to update the **00-INDEX** files in the kernel source tree. **Paul Gortmaker** also had been updating those files, but he posted his patches to the public mailing list instead. That seems to be the standard approach these days, along with creating public git repositories from which Linus and others may pull.

—**ZACK BROWN**

They Said It

Don't go around saying the world owes you a living. The world owes you nothing. It was here first.

—**Mark Twain**

In a networked world, trust is the most important currency.

—**Eric Schmidt**

When you relinquish the desire to control your future, you can have more happiness.

—**Nicole Kidman**

Not a shred of evidence exists in favor of the idea that life is serious.

—**Brendan Gill**

Genius begins great works; labor alone finishes them.

—**Joseph Joubert**

One Tail Just Isn't Enough

```

folkert@muur /var/log/apache/access_log (Sun Feb 26 20:
File Edit View Terminal Tabs Help
crawl-66-249-71-78,googlebot.com - - [26/Feb/2006:20:17:24 +0100] "GET /activite
iten.php HTTP/1.0" 200 2436 "-" "Googlebot/2.1 (+http://www.google.com/bot.html)
"
crawl-66-249-64-37,googlebot.com - - [26/Feb/2006:20:23:23 +0100] "GET /contact.
php HTTP/1.0" 200 1910 "-" "Googlebot/2.1 (+http://www.google.com/bot.html)"
crawl-66-249-71-18,googlebot.com - - [26/Feb/2006:20:24:58 +0100] "GET /overig.p
hp HTTP/1.0" 200 2030 "-" "Googlebot/2.1 (+http://www.google.com/bot.html)"
crawl-66-249-71-18,googlebot.com - - [26/Feb/2006:20:41:40 +0100] "GET /visie.ph
p HTTP/1.0" 200 1561 "-" "Googlebot/2.1 (+http://www.google.com/bot.html)"
64.173.224.240 - - [26/Feb/2006:20:47:55 +0100] "GET / HTTP/1.0" 200 12432 "-" "
-"
[00] /var/log/apache/access_log 13MB - Sun Feb 26 20:47:55 2006
2.176.167]
Feb 26 20:45:36 muur spamd[21966]: clean message (0.0/5.0) for root:99 in 1.8 se
conds, 34848 bytes.
Feb 26 20:45:36 muur spamd[21966]: result: . 0 - RCVD_BY_IP scantime=1.8,size=3
4848,mid=<200602262043.22463.jesper.juhl@gmail.com>,autolearn=failed
Feb 26 20:45:37 muur mail4usscan[4156]: Scanning mail by jesper.juhl@gmail.com
Feb 26 20:45:38 muur mail4usscan[4175]: Message by jesper.juhl@gmail.com is clea
n
Feb 26 20:45:39 muur postfix/local[4105]: 0E1DE1CA368: to=<folkert@vanheusden.co
m>, relay=local, delay=5, status=sent (delivered to command: /usr/bin/procmail)
Feb 26 20:45:39 muur postfix/qmgr[21989]: 0E1DE1CA368: removed
[01] /var/log/mail/mail.log 15MB - Sun Feb 26 20:45:39 2006

```

Figure 1. Two files at once is just the beginning; multitail is awesome (screenshot from <http://www.vanheusden.com>).

Although it's difficult for me to look at this piece's title and not think of mutant felines, it doesn't make the statement any less true. If you've ever used the tail command on log files, you'll instantly appreciate multitail. My friend (and *LJ* reader) Nick Danger introduced me to multitail, and I can't believe how useful it is. multitail will "tail" multiple files, split the screen to display them, notify of log file

changes and so on. One of my favorite features is rather than show 100 lines of repeated log, it shows the line only once, and it says, "line repeats 100 times"—simple, but awesome.

multitail has more features than I could list on this page, but chances are if you've ever wished you could do something with log files, multitail does it. Check it out at <http://www.vanheusden.com/multitail>.—**SHAWN POWERS**

Non-Linux FOSS: libnotify, OS X Style

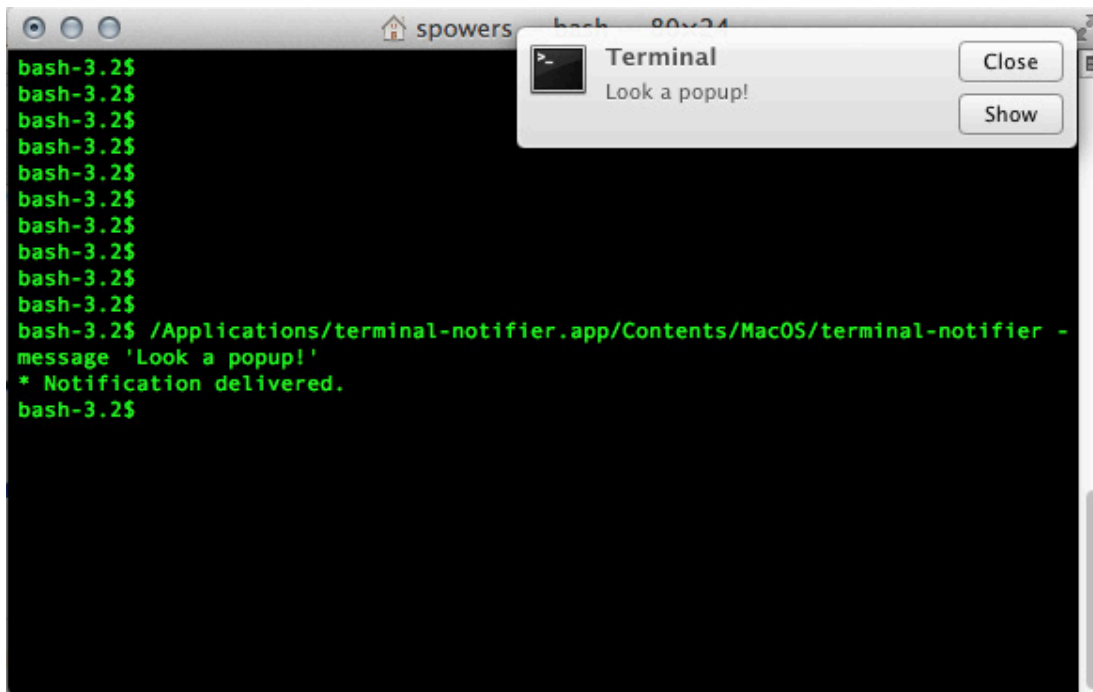


Figure 1. A quick command-line-ninja move creates a pop-up.

One of the things I dislike about using Irssi in a terminal window on OS X is that I often miss the screen flash when someone mentions my name in IRC. With some fancy SSH tunneling (maybe more on that some other issue) and a really cool pop-up notification tool, if someone mentions my name, I can't miss it.

`terminal-notifier` is a command-line tool for creating OS X-native user notifications. It doesn't rewrite the concept of pop-ups; instead, it gives us nerds a way to add pop-ups

until dismissed, but you can change that in the notification settings in OS X's preferences.

If you like pop-up notifications like `libnotify`, but find yourself on a Macintosh machine more often than not, `terminal-notifier` might be as useful for you as it is for me at my day job. Plus, now you know that if you mention my name in IRC during the workday, you'll make a window pop up on my screen! Get it at <https://github.com/alloy/terminal-notifier>.—**SHAWN POWERS**

to scripts (Figure 1). Because it uses the native notification system, it's easy to modify what sort of pop-up appears. I prefer the kind that doesn't go away

Android Poll

We recently asked our readers about their Android usage, and as predicted, most of our readers own an Android phone, tablet or other Android device. Also not surprising is that our readers are mostly up to date, with the majority of users running the Jelly Bean release. 86% are loyal to Android and have not jumped ship in favor of another platform. E-books also are popular with our readers, but E Ink vs. a backlit color display is a toss up. Read on to see the full results, and as always, thanks for participating!

Do you own an Android smartphone?

- Yes: 80%
- No: 20%

Do you own an Android tablet?

- Yes: 56%
- No: 44%

Do you own a non-smartphone/tablet Android device?

- Yes: 56%
- No: 44%

Which version of Android is running on the majority of your devices?

- Cupcake (1.5): <1%
- Donut (1.6): <1%
- Eclair (2.0/2.1): 1%
- Froyo (2.2): 5%
- Gingerbread (2.3): 15%
- Honeycomb (3.0/3.1/3.2): 2%
- Ice Cream Sandwich (4.0): 22%
- Jelly Bean (4.1/4.2): 54%

Did you switch from an iPhone to an Android phone?

- Yes: 10%
- No: 78%
- NA: 12%

Have you switched away from Android to another platform (iOS and so on)?

- Yes: 6%
- No: 86%
- NA: 8%

Have you already or are you planning to buy an Ouya?

- Yes: 11%
- No: 89%

Which is worse for developers?

- iOS's walled garden: 68%
- Android's version fragmentation: 32%

Do you read e-books?

- Yes: 79%
- No: 21%

If you read e-books, which do you prefer?

- An LCD screen: 49%
- E Ink: 50%

Does DRM limit the amount of digital media you purchase?

- Yes: 73%
- No: 27%

What do you do most on your Android device?

- Consume media (audio/video): 34%
- Read/create e-mail: 27%
- Social networking (Twitter, Facebook and so on): 16%

- Play games: 7%
- Talk on the phone: 15%
- Video conference: 1%

Does your device's battery last all day?

- Always: 28%
- Never: 12%
- Usually: 43%
- If I don't use it a lot during the day: 17%

Have you rooted your Android device(s)?

- Yes: 47%
- No: 53%

—**KATHERINE DRUCKMAN**

Powerful: Rhino



- Rhino M4700/M6700**
- Dell Precision M4700/M6700 w/ Core i7 Quad (8 core)
 - 15.6"-17.3" FHD LED w/ X@1920x1080
 - NVidia Quadro K5000M
 - 750 GB - **1 TB hard drive**
 - Up to 32 GB RAM (1866 MHz)
 - DVD±RW or Blu-ray
 - 802.11a/b/g/n
 - Starts at \$1190
 - E6230, E6330, E6430, E6530 also available

- High performance NVidia 3-D on an FHD RGB/LED
- High performance Core i7 Quad CPUs, 32 GB RAM
- Ultimate configurability — choose your laptop's features
- One year Linux tech support — phone and email
- Three year manufacturer's on-site warranty
- Choice of pre-installed Linux distribution:



Tablet: Raven



- Raven X230/X230 Tablet**
- ThinkPad X230/X230 tablet by Lenovo
 - 12.1" HD LED w/ X@1366x768
 - 2.6-2.9 GHz Core i7
 - Up to 16 GB RAM
 - 750 GB hard drive / 180 GB SSD
 - Pen/finger input to screen, rotation
 - Starts at \$2050
 - T430, T530, W530 also available

Rugged: Tarantula



- Tarantula CF-31**
- Panasonic Toughbook CF-31
 - Fully rugged **MIL-SPEC-810G** tested: drops, dust, moisture & more
 - 13.1" XGA TouchScreen
 - 2.4-2.53 GHz Core i5
 - Up to 8 GB RAM
 - 320-750 GB hard drive / 512 GB SSD
 - CF-19, CF-52, CF-H2 also available

EmperorLinux
...where Linux & laptops converge

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



GIS with GRASS

The major player in the Windows world for GIS programs is the suite of ESRI products. In Linux, we have the package named GRASS (<http://grass.osgeo.org>). GRASS originally was developed by the US Army Construction Engineering Research Laboratories, starting

in 1982. It is used by many large groups, including NASA, NOAA and the National Park Service. In September 2006, management of GRASS was taken over by the GRASS Project Steering Committee, and it now is an official project of the Open Source Geospatial Foundation.

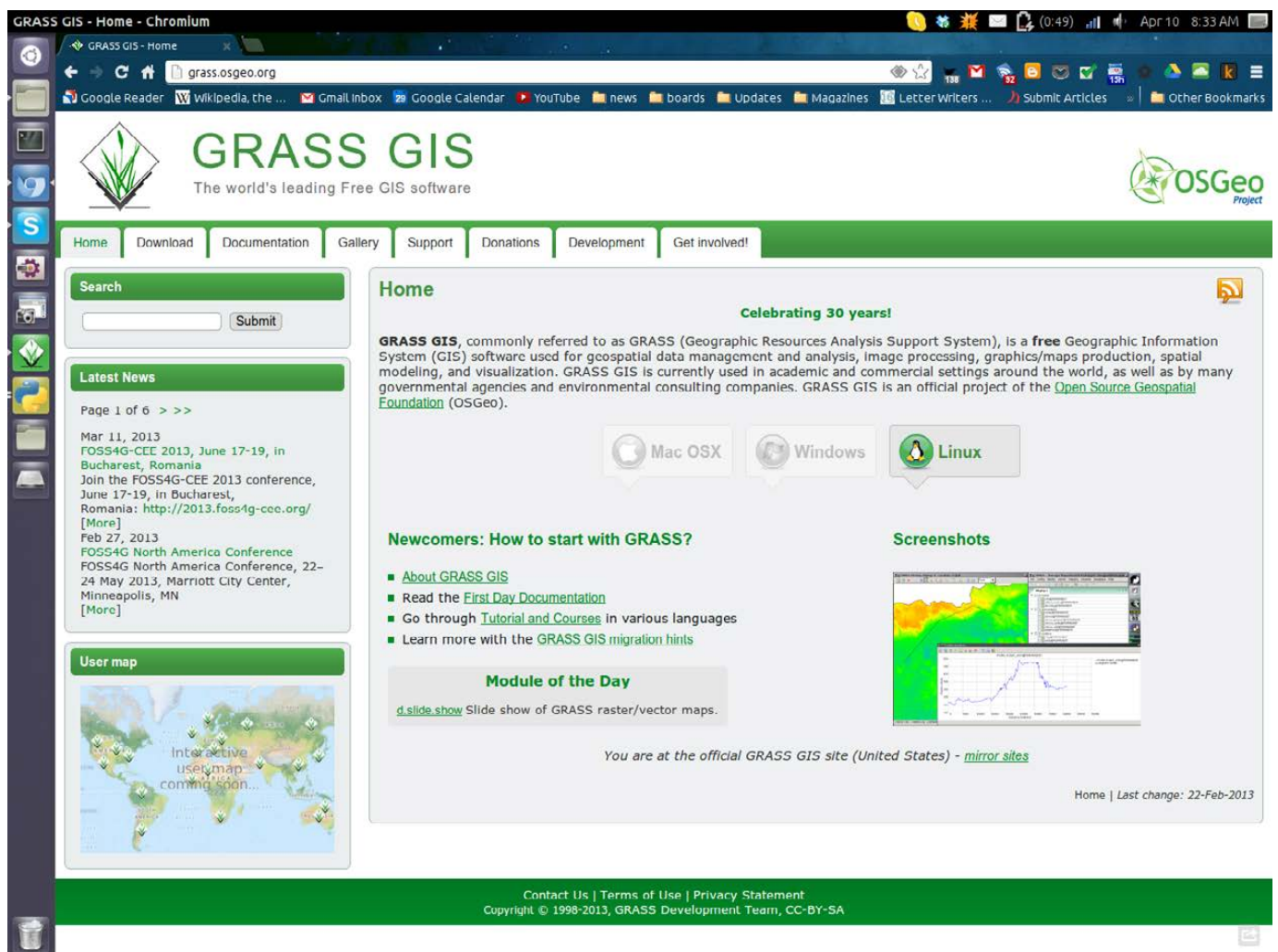


Figure 1. The Main Web Site for the GRASS Project

GRASS may be too much if you just want to do basic GIS tasks. In that case, you may be better served by a program like QGIS. But, if you need to do some serious GIS analysis, GRASS definitely is worth the learning curve. Most distributions should have a set of packages to simplify installation. If you do want the latest source, or need a version of GRASS for

Windows or Mac OS X, you always can go to the main Web site.

When you first start up GRASS, it asks you to set a data directory. The suggestion is ~/grass-dir. Once you select this data directory, you need to set some project information. You can click on the Location Wizard to help set the location information for your project.

Once you set the name and data

The screenshot shows the GRASS GIS website's download page. The browser's address bar displays grass.osgeo.org/download/. The page header includes the GRASS GIS logo and the tagline "The world's leading Free GIS software", along with the OSGeo Project logo. A navigation menu contains links for Home, Download, Documentation, Gallery, Support, Donations, Development, and Get involved!. On the left sidebar, there is a search bar, a "Latest News" section with two entries (dated Mar 11, 2013 and Feb 27, 2013), and a "User map" section showing a world map with location markers. The main content area is titled "Download" and contains the text "Get here your free and full GIS software copy:" followed by three buttons for "Mac OS X", "Windows", and "Linux". Below this, there are sections for "Source code" with links to source code packages, installation guides, and software licenses; "Citing GRASS Software" with instructions and links to citation repositories; and a BibTeX entry for LaTeX users. The footer of the page includes contact information and copyright details: "Copyright © 1998-2013, GRASS Development Team, CC-BY-SA".

Figure 2. The download page provides binaries for Linux, Windows and Mac OS X.



Figure 3. When you initially start GRASS, you need to select a location and a mapset.

directory, you need to select the method for creating a new location. Just to get started, you simply can accept the defaults. To learn how to work with GRASS, you will want to have some data to play with. Sample datasets are available to download from the main GRASS site (<http://grass.osgeo.org/download/sample-data>). Choose one or more of them, download the files and uncompress them into the data directory you set above. These sample datasets then will show up in the “Welcome to GRASS” window when you first start up GRASS. At this point,

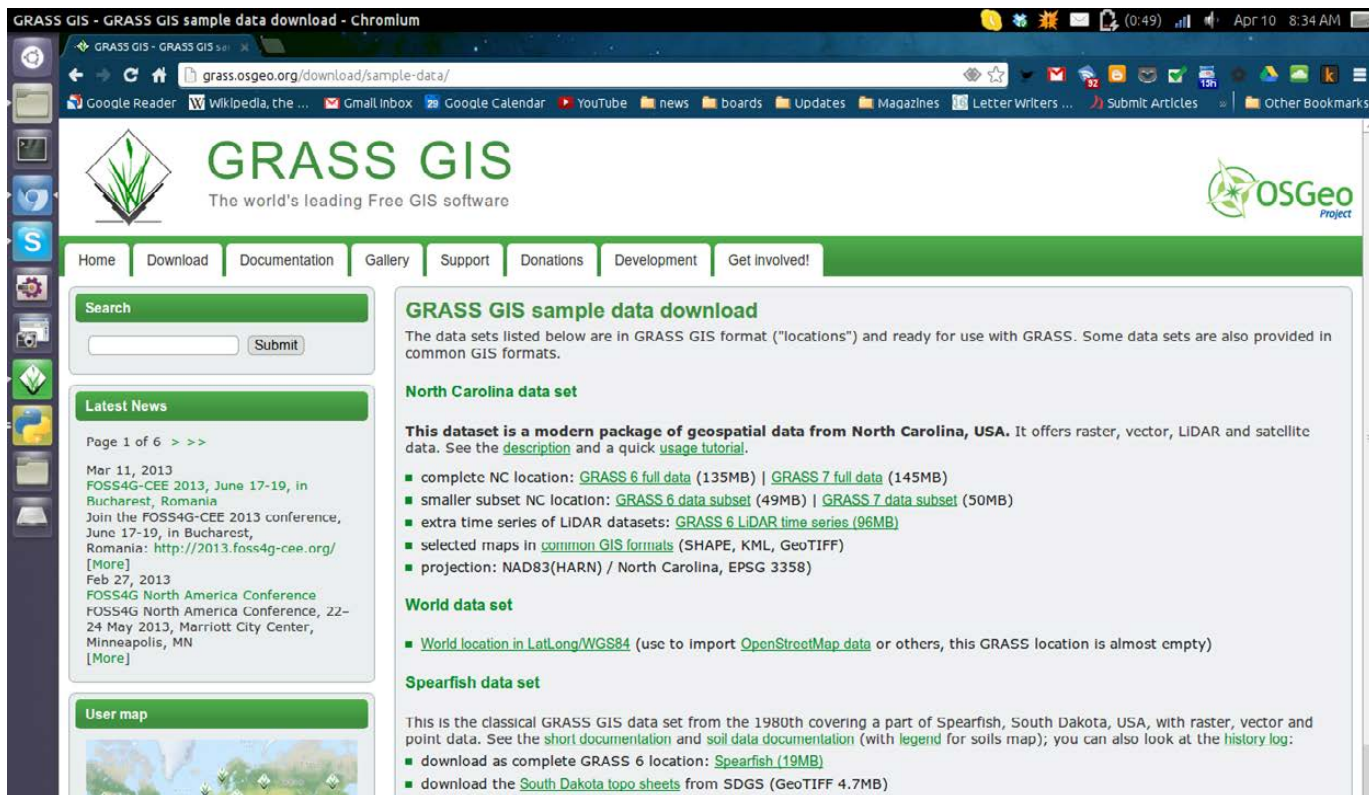


Figure 4. You can use several sample data sets while you are learning to use GRASS.

select one of these datasets. You also need to select a mapset, most usually PERMANENT.

Once GRASS starts up, two windows appear. The first window is a map display, where all of the layers you select will be rendered. The second window is where you select the map layers that you want to apply to the map display window.

To create your first map, click on the "Add raster map layer" button

(the one with a checkerboard and a plus sign), or you can press Ctrl-Shift-R. This will pop up a dialog window where you can select which layer you want to add from the mapset you loaded on startup. In this example, I have loaded the PERMANENT mapset from the spearfish location, and I set the 10m elevation as the first layer of my map.

One of the first things you will

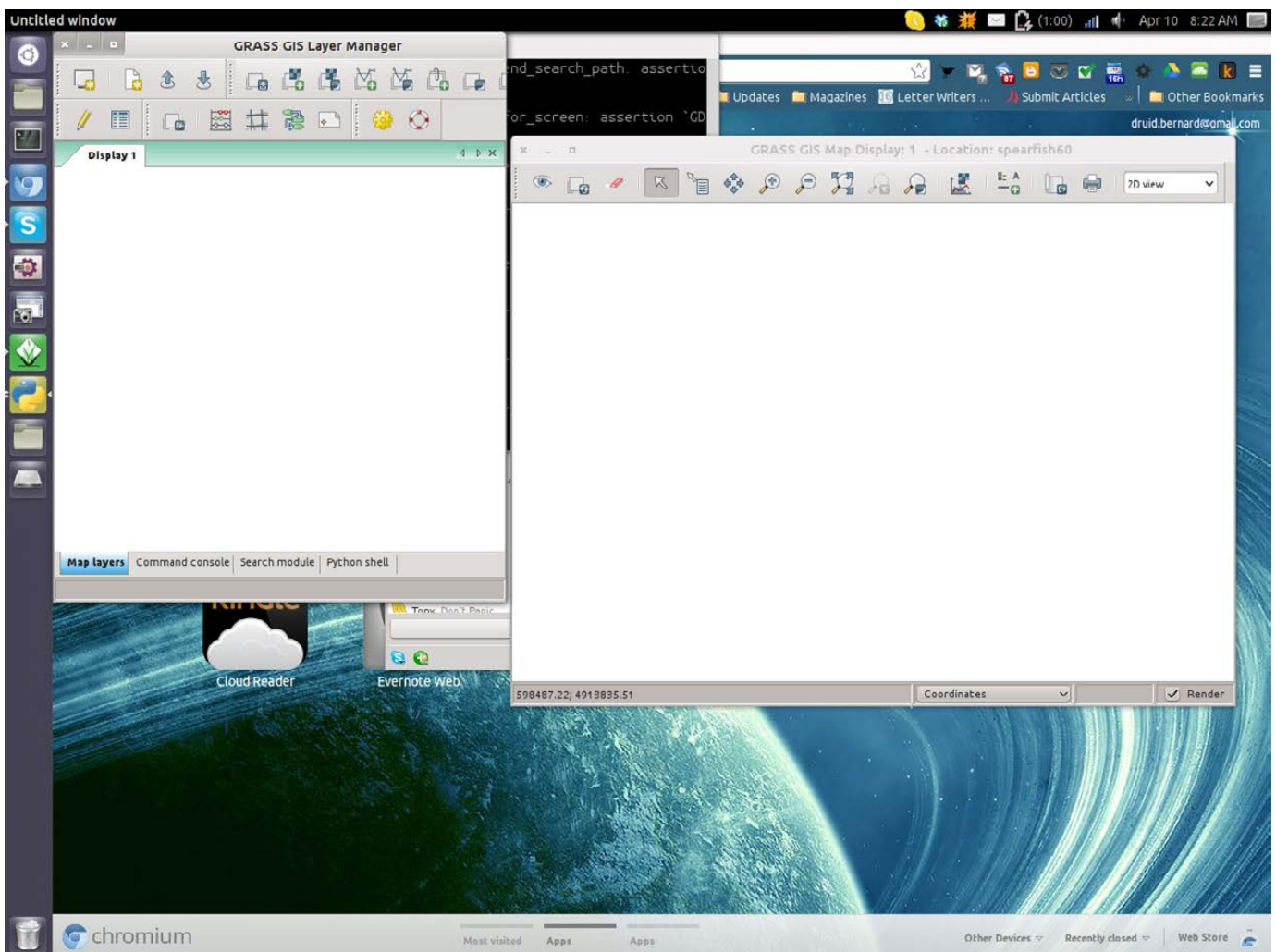


Figure 5. On start up, you will have a map display window and a layers window.

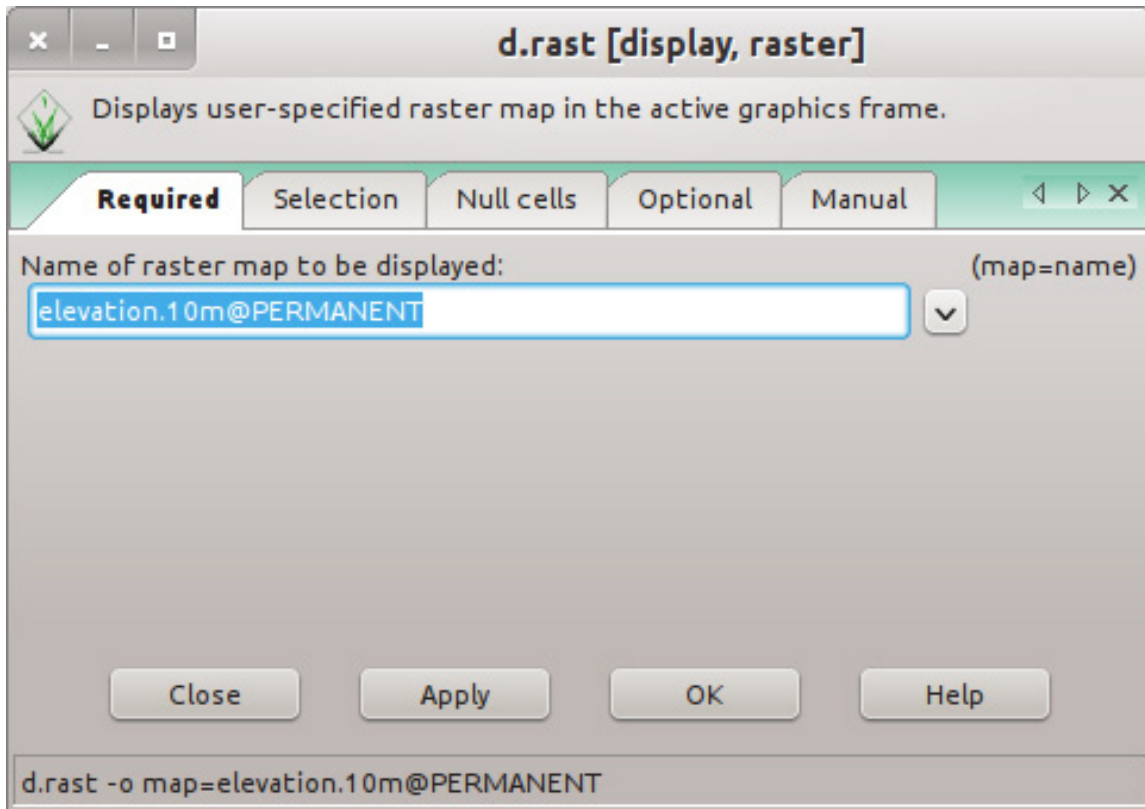


Figure 6. Here you select which layers to load.

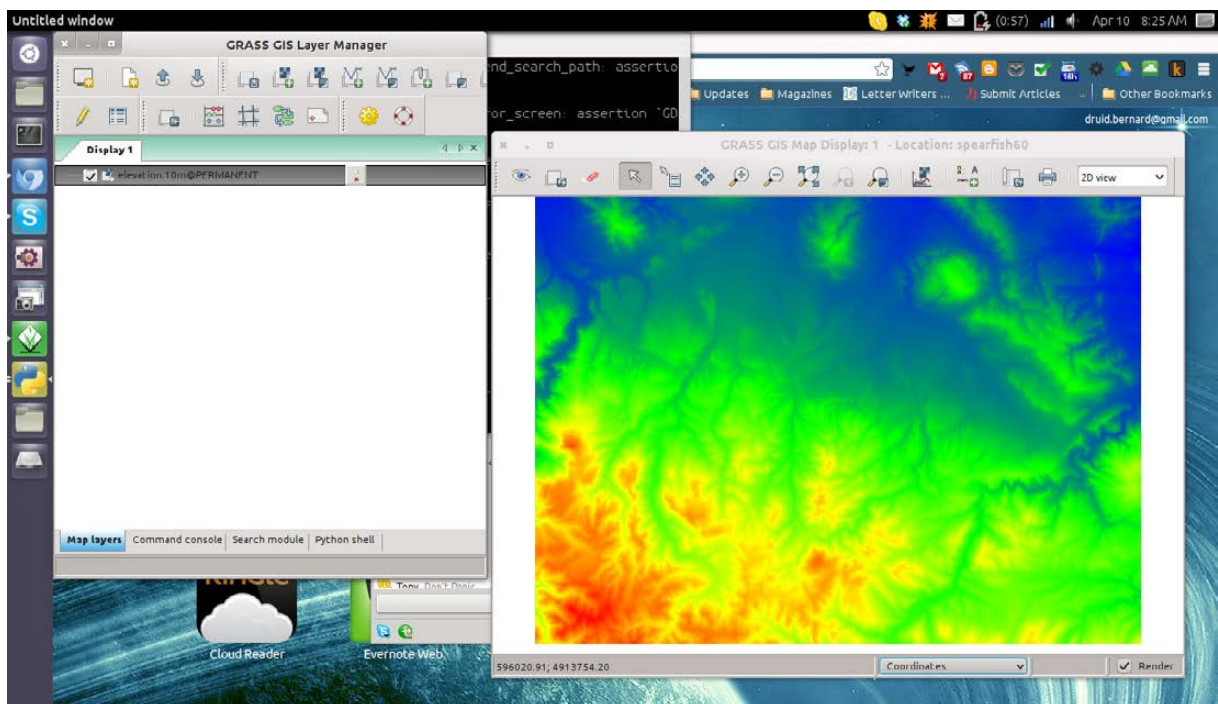


Figure 7. Here you can see the elevation of the land in the spearfish location.

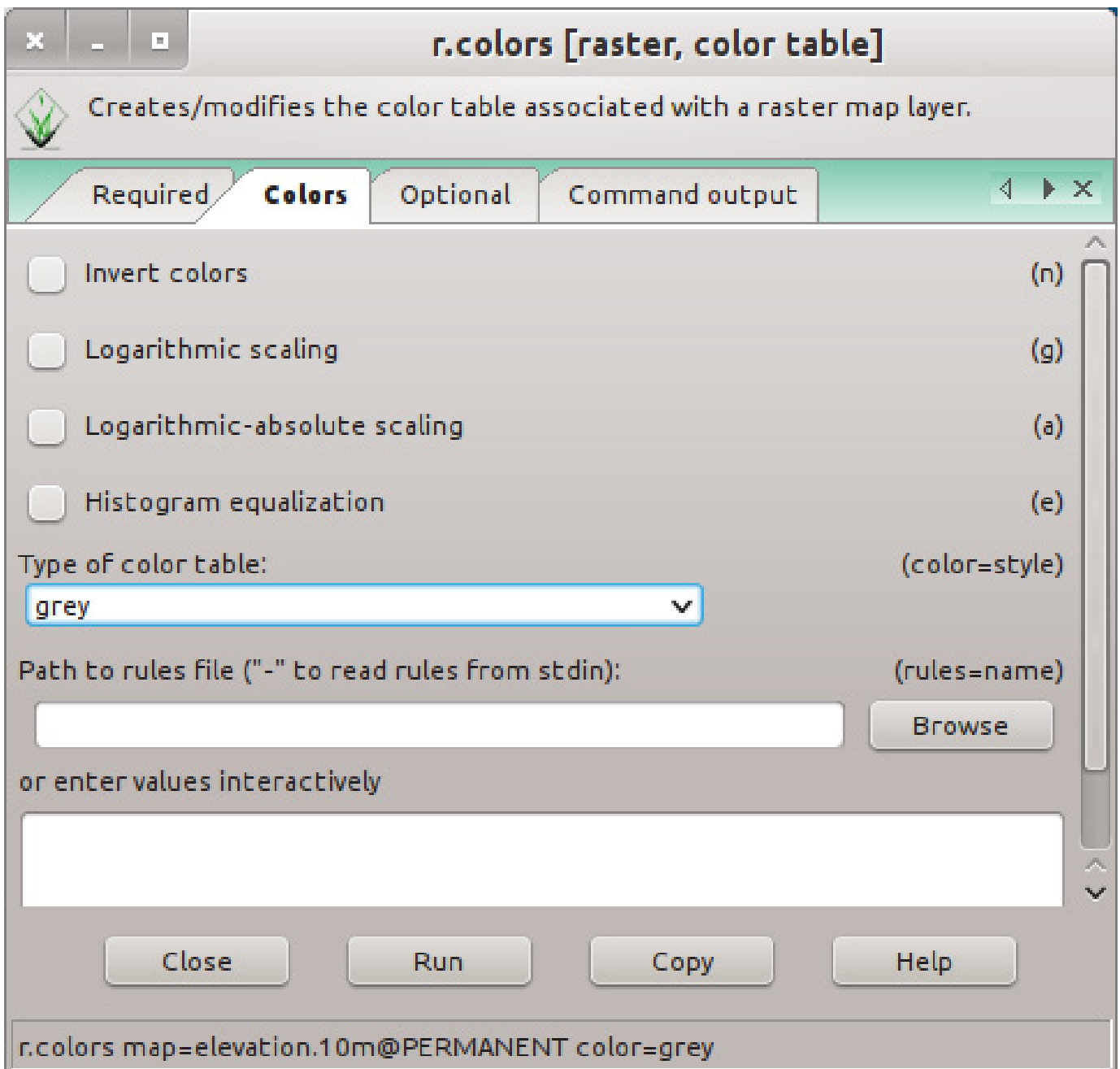


Figure 8. Changing the color map is relatively simple.

want to do is to change the colors used within the map. To do this, right-click on the layer in question, and select “Set color table” from the drop-down menu. You then can change the color table that GRASS

will select from in order to render the layer on your map. To change your layer to grayscale, select “Type of color table:” and select “grey”.

When you click on the run button, you are switched to the “Command

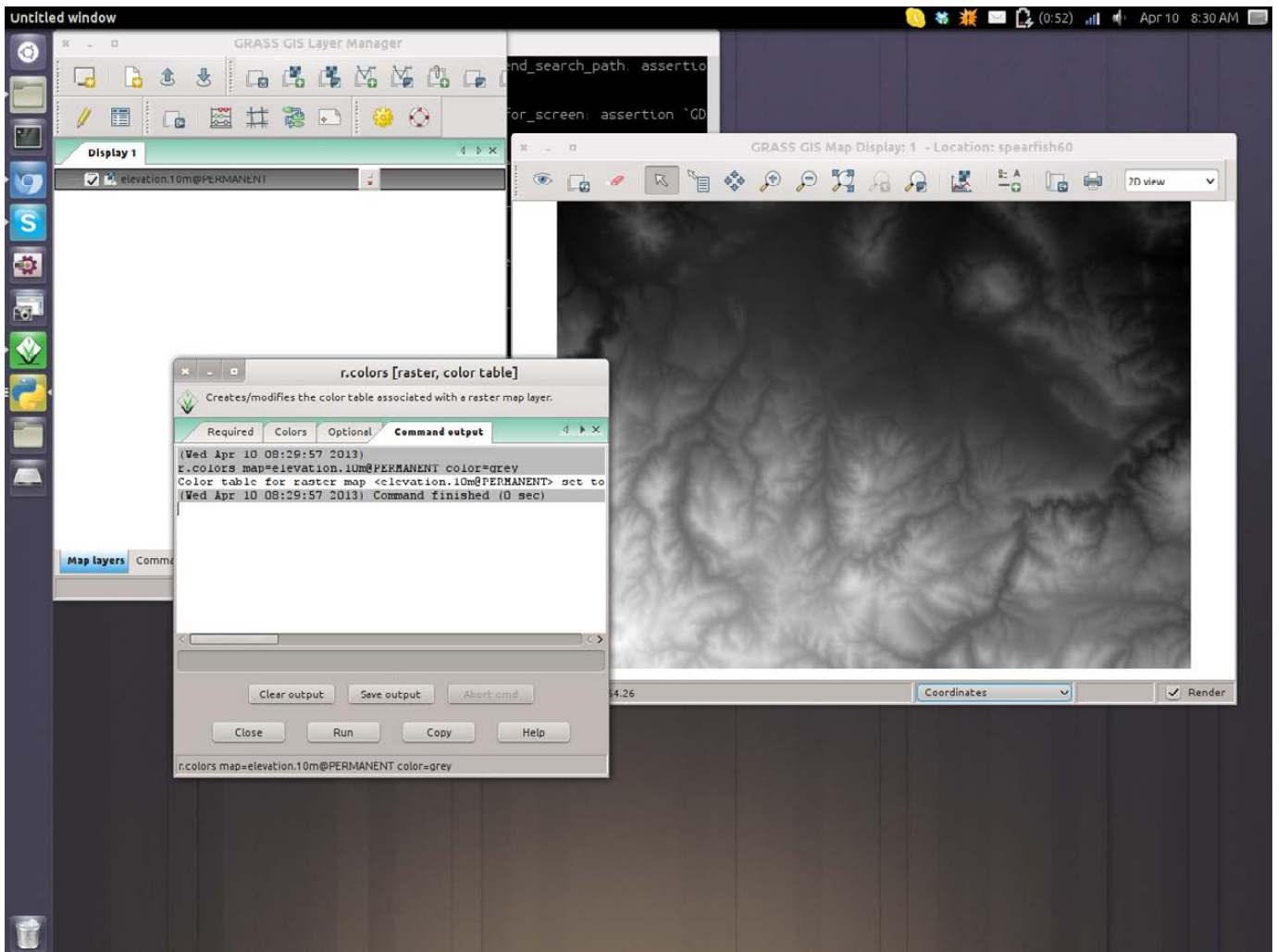


Figure 9. You can see the results of running the color change immediately.

Output” tab where the results from this command are displayed. If you want to see an idea of the spread of the possible values, you can get a histogram by right-clicking the layer and selecting Histogram. If you need more exact numbers, you actually can calculate univariate statistics on the data in the layer. This is done by right-clicking on the layer and selecting “Univariate

raster statistics”.

Adding a second layer allows you to start building up the information being displayed on your map. You need to be careful of which order the layers are in the layer list. They are rendered from the bottom up. This means that layers further up the stack may obscure what is being displayed lower down. You may need to change

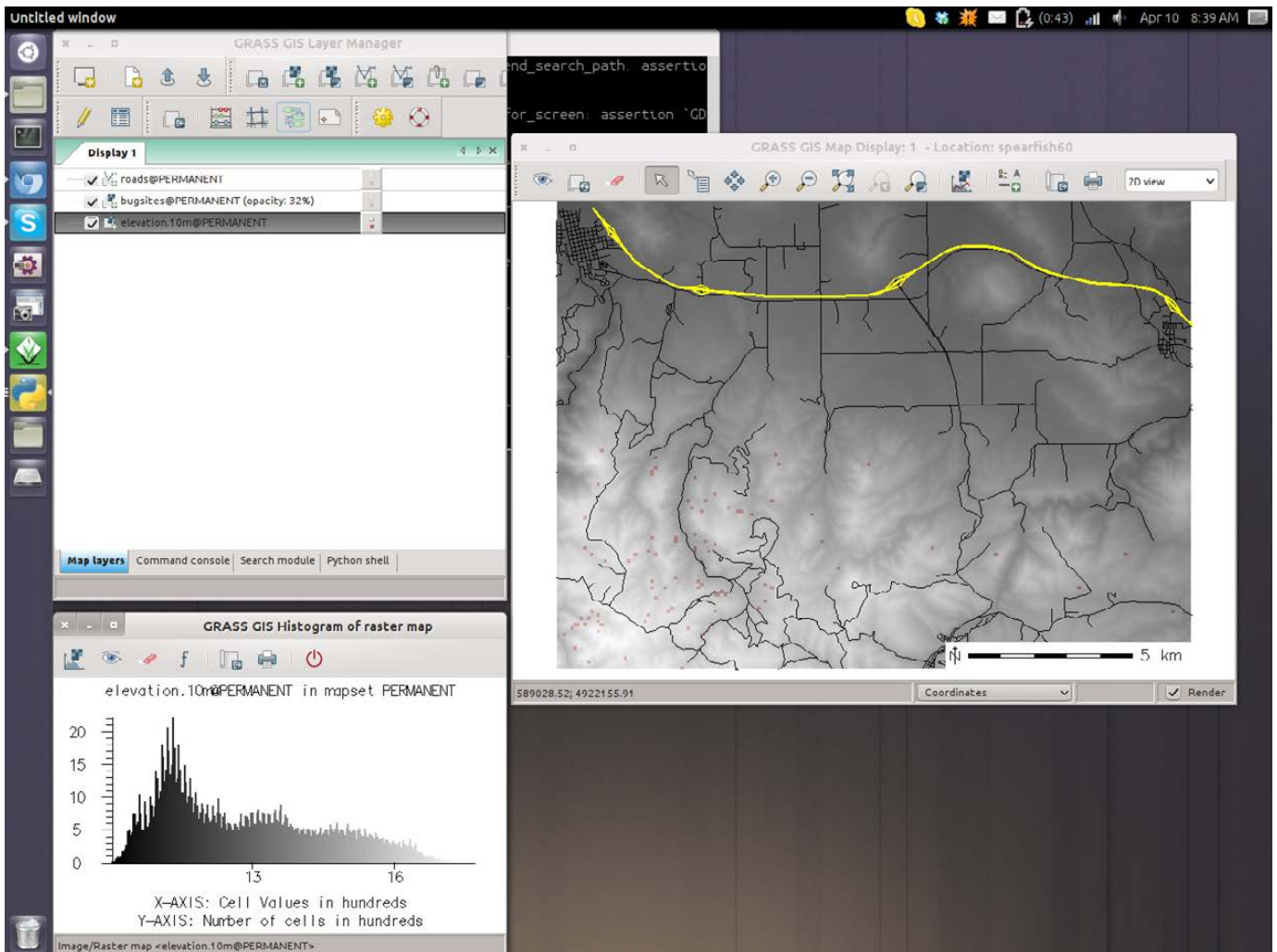


Figure 10. You can get an idea of the spread of values in a raster map by generating a histogram.

the opacity of the upper layers to allow information from the lower layers to show through. Right-click on the layer in question and select “Change opacity level”. You then can set it to an appropriate level so everything you want to see actually is rendered.

The other type of layer that you can add to your map is a vector

layer. In this case, the data is stored as a set of geometrical objects, where each object has some attribute data assigned. With vector layers, the only portions that are rendered are the actual objects. For example, if you add a road layer, you don’t need to worry about opacity, because the roads are small enough not to obstruct

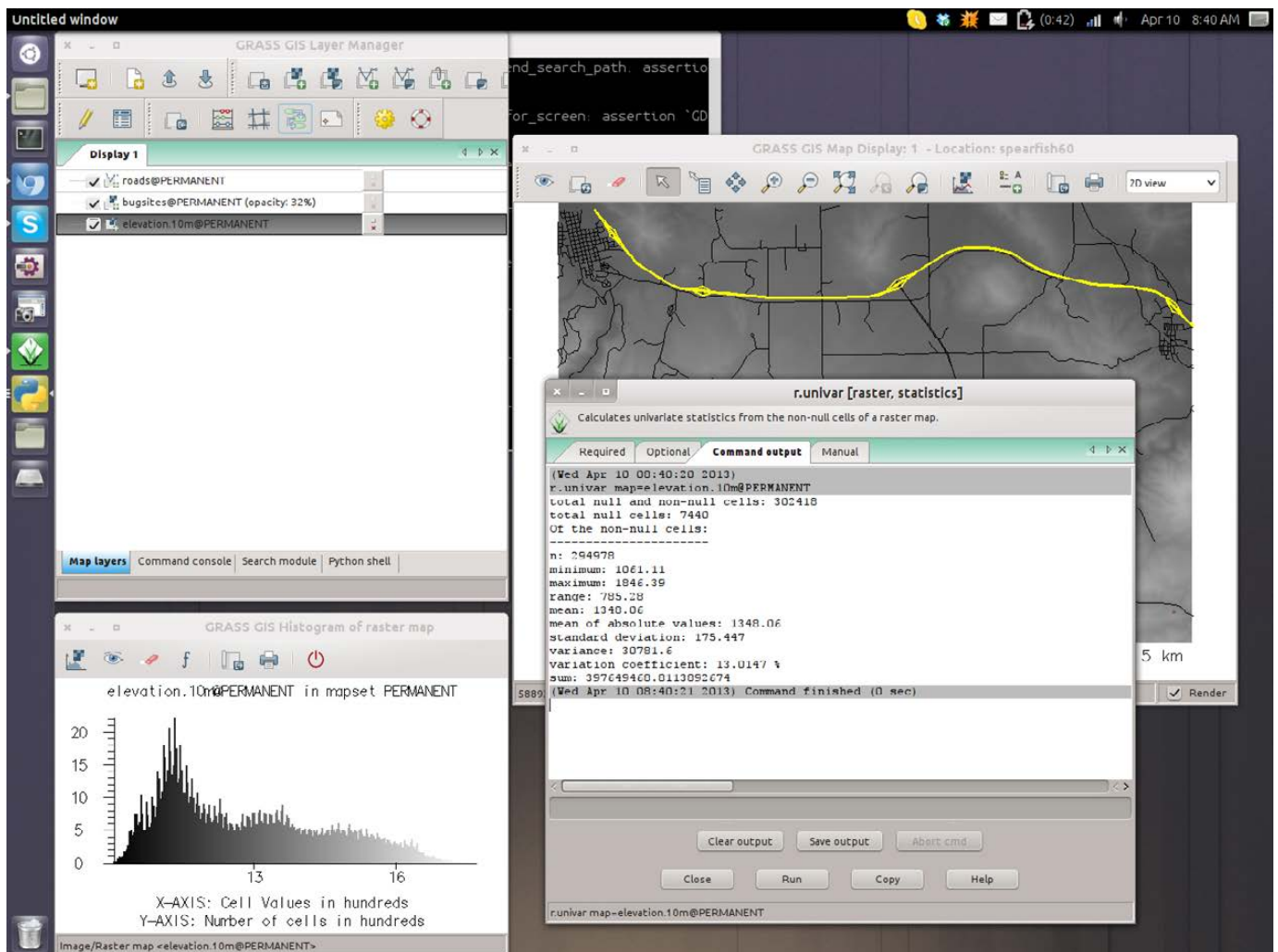


Figure 11. More detailed analysis is available by looking at univariate statistics.

anything on the layers below. You can right-click on that layer and edit the attribute data. You then can select which values for each attribute to display. This can be a more complex selection—for example, selecting those values between an upper and lower bound or selecting only those values that match some other criterion.

You can change display properties

for the objects by right-clicking and selecting Properties. For the road layer, you can set properties like line width, line color and what symbols to use for point elements.

You can add extra elements that you normally see on maps by selecting the “Add map elements” button on the main map display. This opens a drop-down box where you can select extra elements to

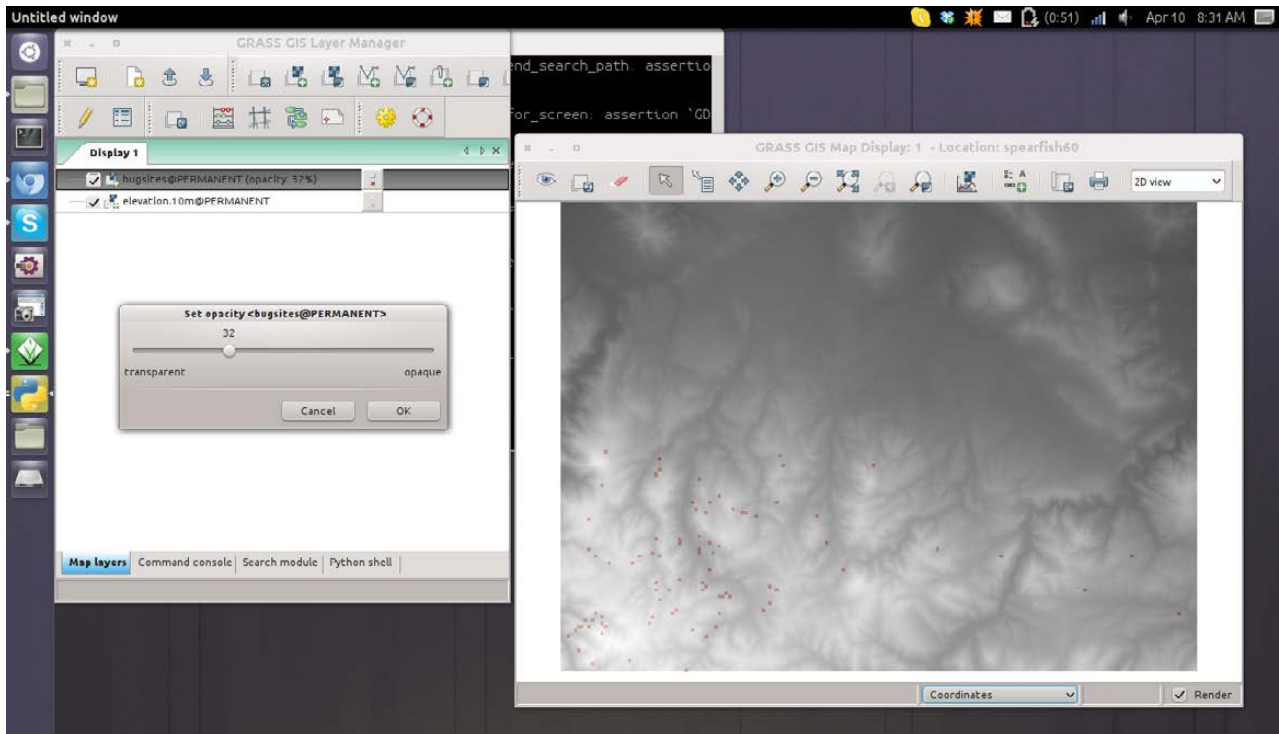


Figure 12. You may need to change the opacity of layers once you start stacking them.

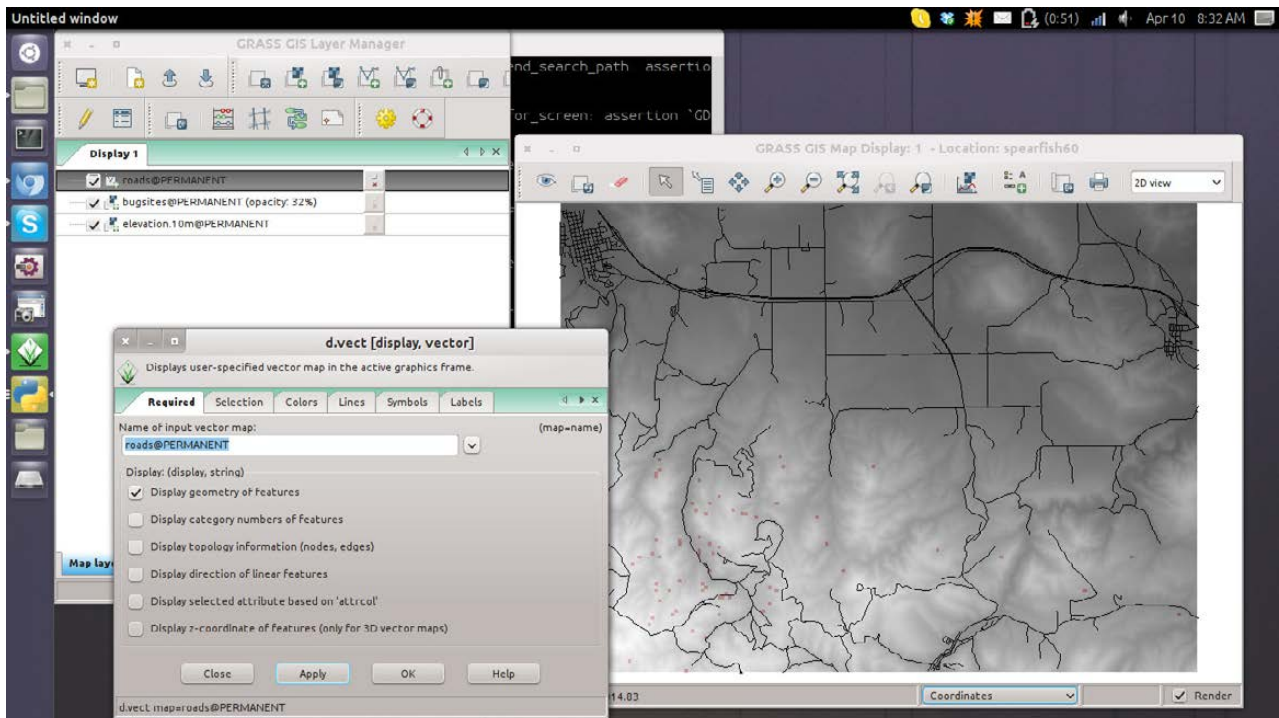


Figure 13. You also can load vector layers.

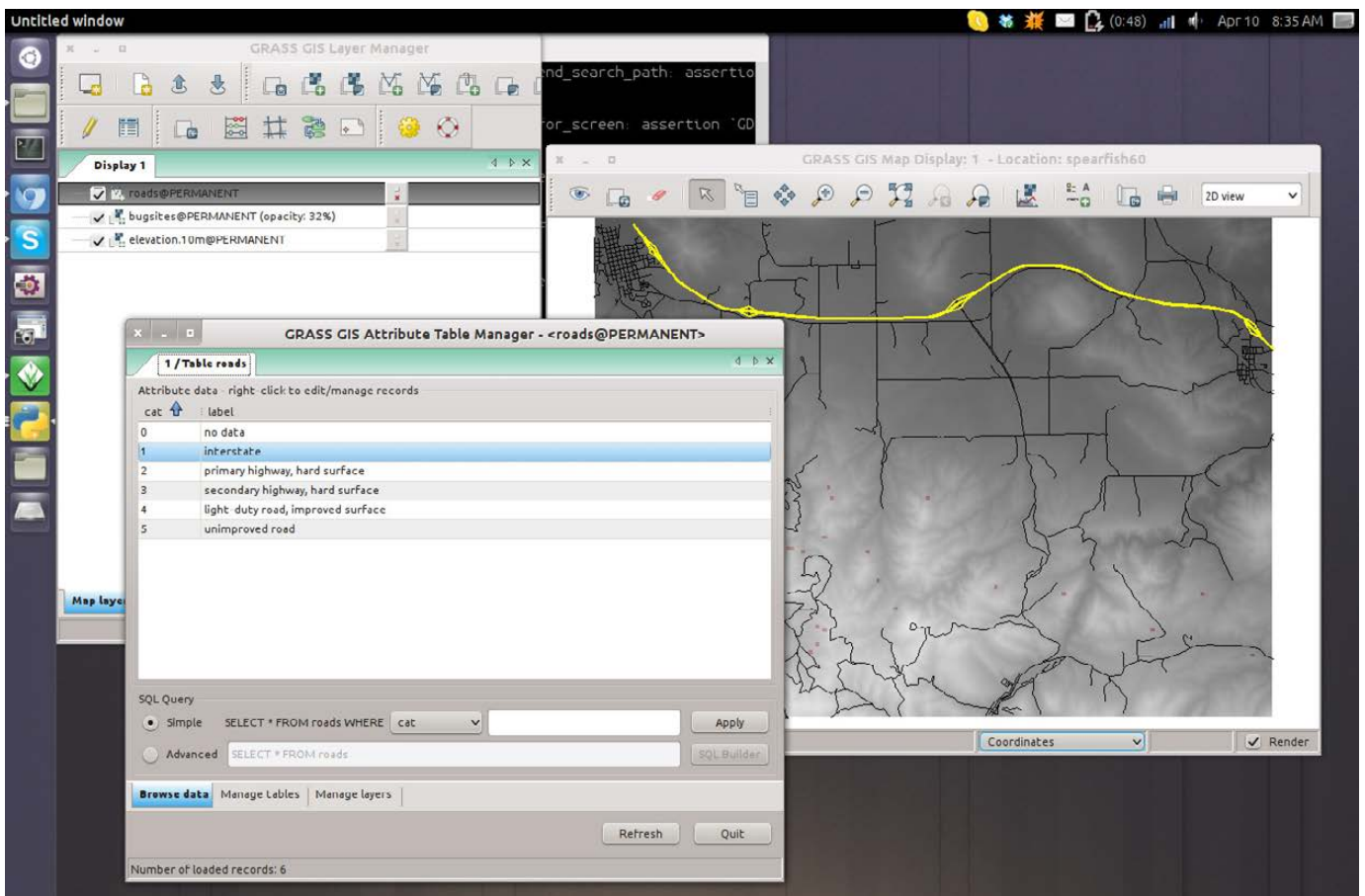


Figure 14. You can highlight elements in a vector layer.

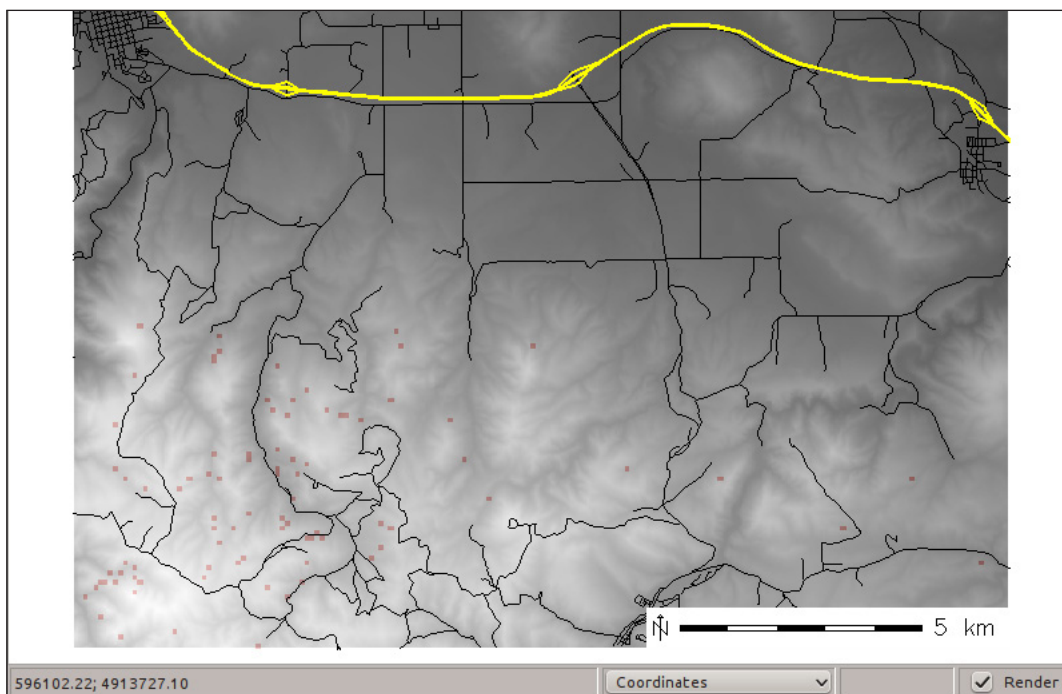


Figure 15. You can add extra elements to your map, like a scale bar.

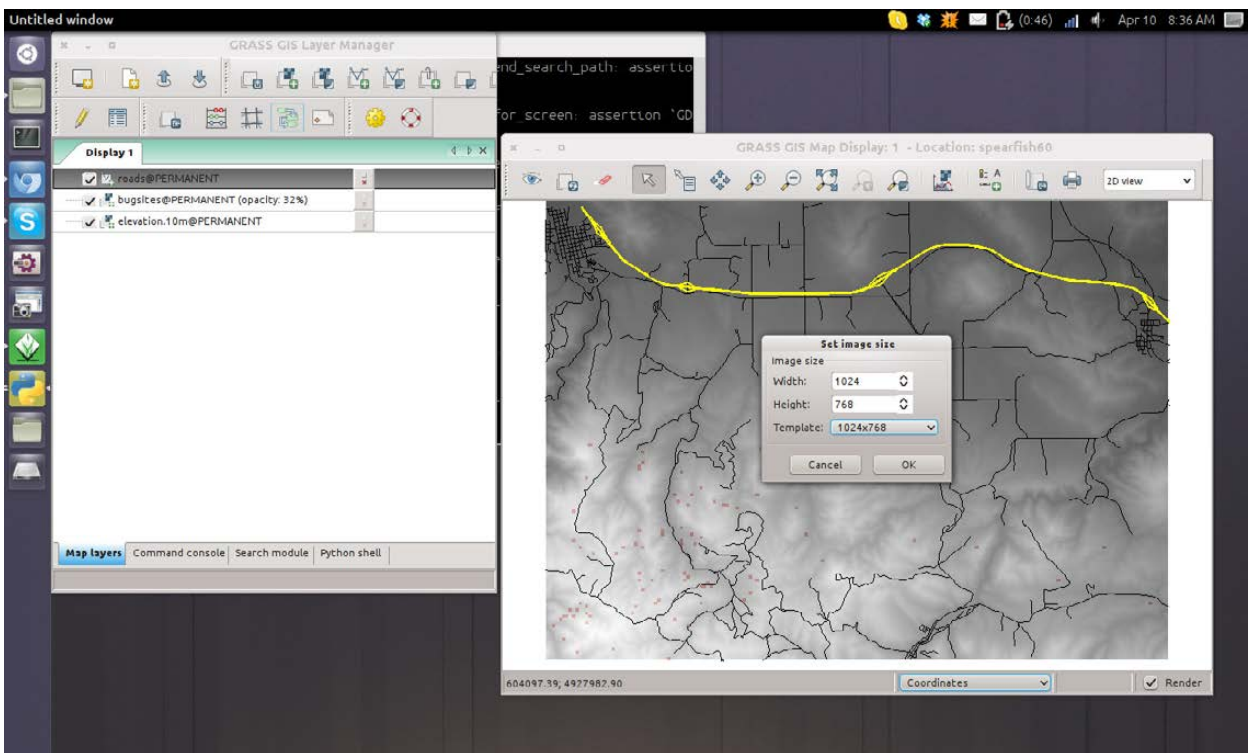


Figure 16. When saving a map, you need to select the output size.

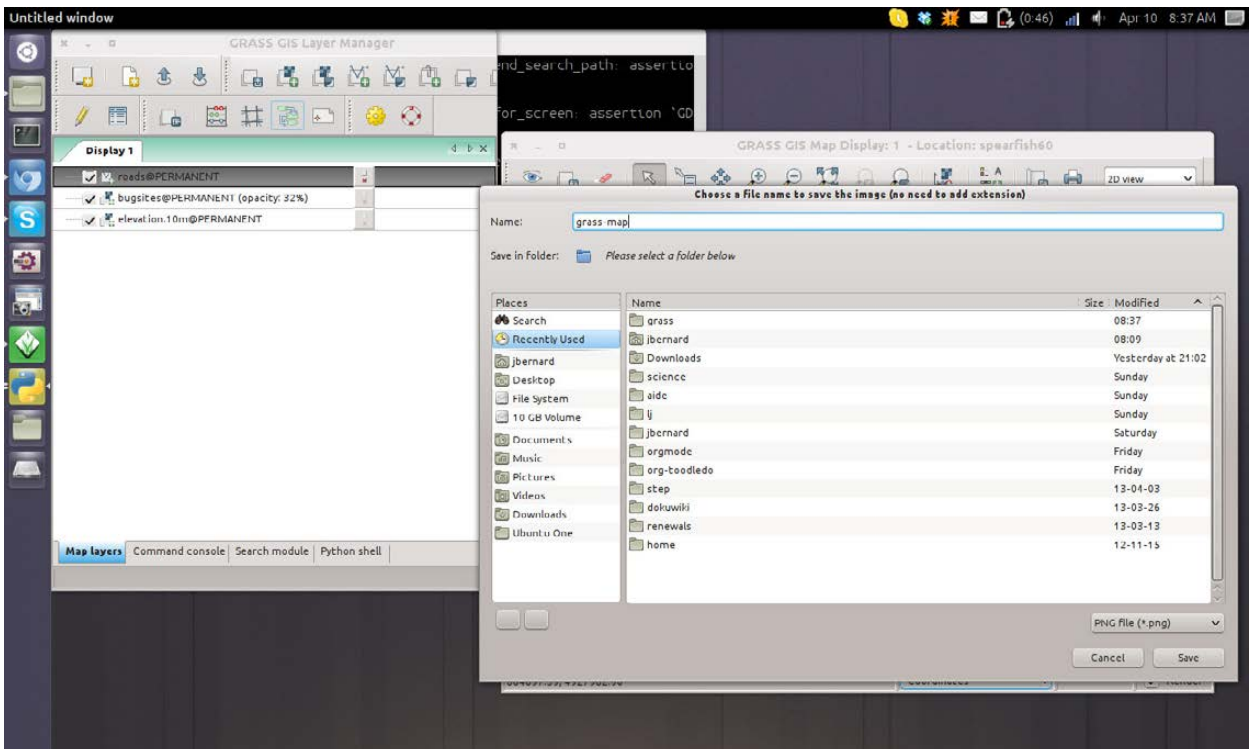


Figure 17. You can select the filename and file format when you save your map.

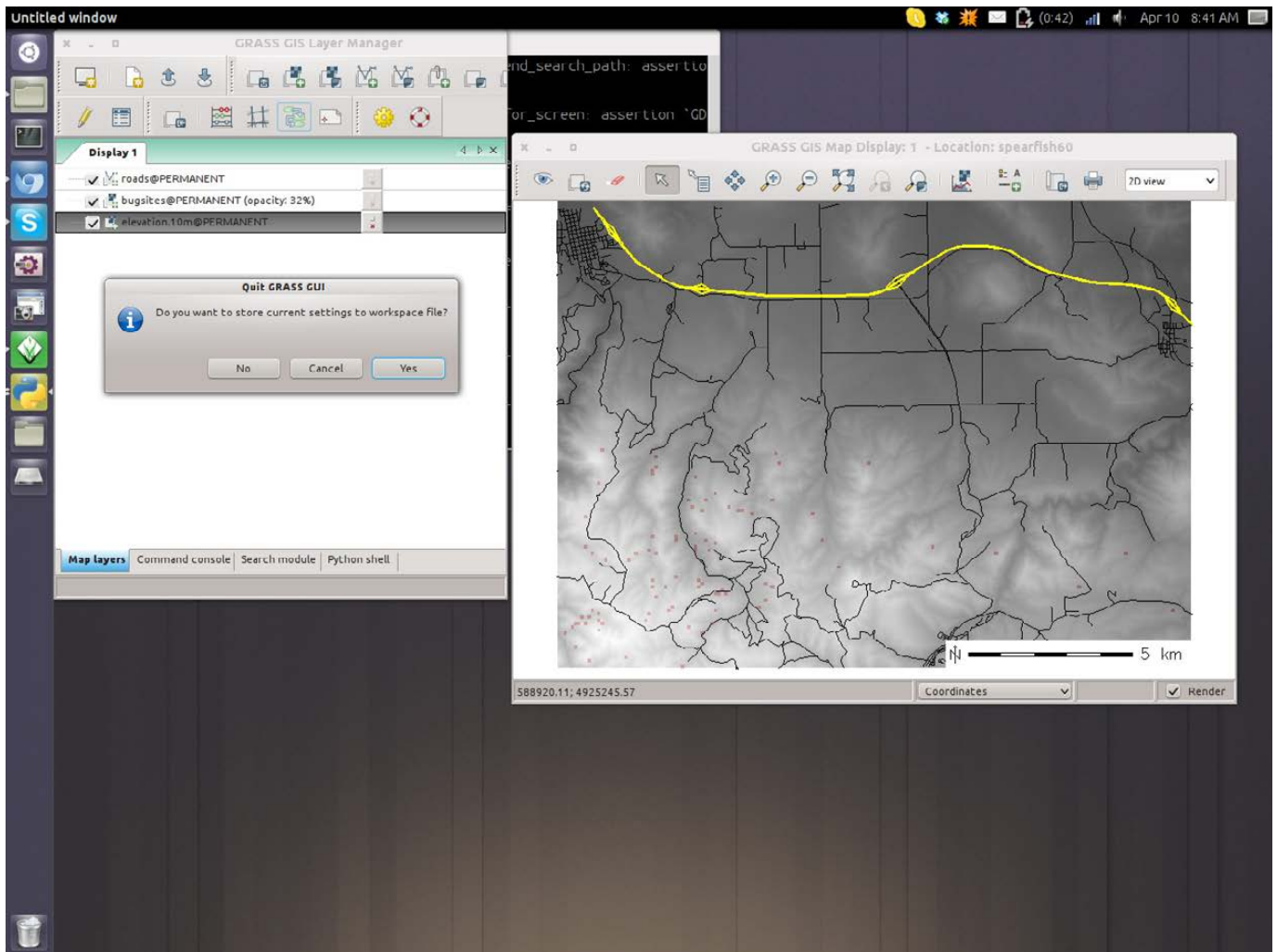


Figure 18. Don't forget to quit when you're done.

add. These include scalebars, North arrows, legends and text areas. You can click and drag these elements and place them where they need to be on your map.

Once you have the layout the way you want, you need to save a final copy so you don't lose all of your work. To do so, click the "Save display to graphic file" button on the main map display. The first step

is to choose the output size for the map. Then you can select the filename and the file format.

Hopefully, this article introduces you to enough of GRASS to induce you to try it out. If it's good enough for the US Army, it's good enough for me. It should be powerful enough to handle any GIS task you have.

—JOEY BERNARD

1&1 DYNAMIC CLOUD SERVER PRICE CONTROL



AS LOW AS
\$0.06
PER HOUR*
PLUS \$60 OFF
FIRST MONTH

- ✓ **COMPLETE COST CONTROL**
 - Full transparency with accurate hourly billing
 - Parallels® Plesk Panel 11 included, with unlimited domains
 - Sleep mode feature
- ✓ **FULL ROOT ACCESS**
 - The complete functionality of a root server with dedicated resources
- ✓ **MAXIMUM FLEXIBILITY**
 - Configure the Processor Cores, RAM and Hard Disk Space
 - Add up to 99 virtual machines
- ✓ **FAIL-SAFE SECURITY**
 - Redundant storage and mirrored processing units reliably protect your server



DOMAINS | E-MAIL | WEB HOSTING | eCOMMERCE | SERVERS

Call 1 (877) 461-2631 or buy online

1and1.com

*Other terms and conditions may apply. Visit www.1and1.com for full promotional offer details. Customer is billed monthly for minimum configuration (\$0.06/ hour * 720 hours = \$43.20/ month minimum). A \$60 credit (valid only for the first month) will be applied to your first month of service. Program and pricing specifications and availability subject to change without notice. 1&1 and the 1&1 logo are trademarks of 1&1 Internet, all other trademarks are the property of their respective owners. © 2013 1&1 Internet. All rights reserved.

Weechat, Irssi's Little Brother



It may not be fair to call Weechat the little brother of Irssi, but in my short introduction to it, that's what it felt like. If Weechat didn't seem quite as powerful as Irssi to me, I definitely can say that it is better-looking out of the box. So, little brother has one thing going for him!

The other day, I was tweeting with Janne Jokitalo about the sorts of things two nerds tweet about: command-line editors and command-line chat clients. I mentioned Irssi in a screen, and he mentioned Weechat. I'm glad he did! Right out of the box, Weechat

```

Welcome to the social channel for Linux Journal - The Original Magazine of the Li>>
| subdriven sunfun t_s_o tabasko teratoma @ChanServ
| torrancew tyldis VadtecWk yesimnathan] @EvilBill
07:56:32 -- | Channel #linuxjournal: 91 nicks (4 ops, 0 @greenfly
| halfops, 1 voice, 86 normals) @JustinRyan
07:56:32 -- | URL for #linuxjournal: +gb31
| http://www.linuxjournal.com adrianhannah
07:56:34 -- | Mode #linuxjournal [+Ccnt] Aegil
07:56:34 -- | Channel created on Mon, 31 Mar 2008 AlisonChaiken
| 17:16:17 Arcticfoxxi0
07:56:58 shawnp0w3rs | Trying out weechat... I like the list of ArdRigh
| users on the side. avis
07:57:06 shawnp0w3rs | Let's see... bbrandt
07:57:08 shawnp0w3rs | !tea bettaa
07:57:08 * | gb31 announces "It's four o'clock Bryanstein
| somewhere!" and grabs the Ceylon. caveat-
07:57:16 shawnp0w3rs | Yep, tastes the same as in irssi cbmuser
07:57:26 shawnp0w3rs | !coffee classix
07:57:26 * | gb31 offers a mocha latte, half fat, extra cmdshftn
| foam. cod3r
07:57:31 shawnp0w3rs | !iv ColonelPanic001
07:57:32 * | gb31 scrubs up, snaps his glove, and constant
| inserts an espresso IV drip. Corey
07:57:43 shawnp0w3rs | Yep, this seems about right. Good morning datacrusher
| everyone. :) daveyates ++

[07:57] [2] [irc/freenode] 2:#linuxjournal(+Ccnt){91} [Act: 1]
[shawnp0w3rs(i)]

```

Figure 1. The *Linux Journal* IRC bot works in Weechat, so I was happy.

does some things I've never been able to get Irssi to do well. First off, it has a list of users docked to the right side of the terminal (Figure 1). I always liked that feature in the GUI client X-Chat, but I couldn't get it to work well in Irssi. I also think the look and feel is far more friendly than that of Irssi. Yes, with the help of Kyle Rankin, I've been able to tweak Irssi into the perfect chatting machine, but Weechat seems to have a more gentle learning curve.

It supports IRC and Jabber right now, but the Web site claims more protocols are coming. Weechat is probably already in your distro's repository, so install it, and give it a whirl. You'll get all the geek creed of Irssi with some fancy interface additions! Due to its focus on usability and its roots in hard-core nerd-dom on the command line, Weechat is this month's Editors' Choice selection. Check it out at <http://www.weechat.org>.

—**SHAWN POWERS**

LINUX JOURNAL WEBCAST

Tuesday, June 18
at 10 am PDT

LINUX JOURNAL PRESENTS:

How to Build an Optimal Hadoop Cluster to Store and Maintain Unlimited Amounts of Data Using Microservers

SPEAKERS:

Hortonworks: AJAY SINGH
Director, Technical Channels

AMD | SeaMicro: MICHAEL PARTRIDGE
Senior Member Technical Staff

Realizing the promise of Apache® Hadoop® requires the effective deployment of compute, memory, storage and networking to achieve optimal results. With its flexibility and multitude of options, it is easy to over or under provision the server infrastructure, resulting in poor performance and high TCO. Join us for an in-depth, technical discussion with industry experts from leading Hadoop and server companies who will provide insights into the key considerations for designing and deploying an optimal Hadoop cluster.

Some of key questions to be discussed are:

- What is the "typical" Hadoop cluster and what should be installed on the different machine types?
- Why should you consider the typical workload patterns when making your hardware decisions?
- Are all microservers created equal for Hadoop deployments?
- How do I plan for expansion if I require more compute, memory, storage or networking?

REGISTER NOW: <http://www.linuxjournal.com/AMDHadoop>

Presented by

LINUX
JOURNAL™

AMD 



REUVEN M.
LERNER

Unicode

Support legacy data and users better by understanding Unicode.

Let's give credit where credit's due: Unicode is a brilliant invention that makes life easier for millions—even billions—of people on our planet. At the same time, dealing with Unicode, as well as the various encoding systems that preceded it, can be an incredibly painful and frustrating experience. I've been dealing with some Unicode-related frustrations of my own in recent days, so I thought this might be a good time to revisit a topic that every modern software developer, and especially every Web developer, should understand.

In case you don't know what Unicode is, or how it affects you, consider this: in C and in older versions of languages like Python and Ruby, a string is nothing more than a bunch of bytes. There's no rhyme or reason to it; you can read whatever data you want into a string, and the language will be fine with it. For example, if I fire up iPython (which uses Python 2.7), I can read a JPEG image into a string:

```
s = open('Downloads/test.jpg').read()
```

Most of the time, you use strings not to hold JPEG images, but rather to hold text. If your text is all in English, you're in luck, because all the characters used by the English language are defined in ASCII, a standard that defines 128 different characters, each with a unique number. Thus, character 65 is uppercase A, and the space character is number 32. ASCII is great, and it works just fine—until you want to start using languages other than English.

The problem is most languages require characters that are not used in English, and that aren't defined in ASCII. This means if you want to write words in French, let alone in Arabic or Chinese, you won't have a way to represent characters using ASCII.

A solution for alphabetic languages was a set of ISO standards (ISO 8859-*), which took advantage of the fact that ASCII uses only 7 bits, but that data is transmitted with 8 bits. If you can take advantage of all 8 bits, you double the number

of available characters, from 128 to 256. This is more than enough for languages with a defined alphabet. Thus, Western European languages were defined in ISO-8859-1, Hebrew in ISO-8859-8 and so forth. Moreover, these ISO standards were meant to make it possible to mix the “foreign” language with English. Thus, you could have a document with English and French or English and Arabic. ASCII characters retained their original values, and the non-ASCII characters were defined in the upper 128.

But, what happens when you want to have a document that contains English, Arabic and French? In the ISO-8859 family of standards, there wasn't any way to accomplish this. The same number that was used to describe an accented character in French also would be used to describe a character in Arabic. The program displaying the text in question was responsible for deciding which language and, thus, which characters, would be displayed. A document written in Russian (ISO 8859-5) but displayed by a program expecting Hebrew (ISO 8859-8) would show Hebrew characters, or rather, gibberish.

Things are even worse if you're working with non-alphabetic languages, such as Chinese. Even if

you would like to use the upper-128 characters to write Chinese, you would be forced to choose from a tiny percentage of the characters that are necessary to use the language. Clearly, something else would be necessary, and indeed, the Chinese (as well as Japanese) invented their own systems for storing text on computers, which were completely incompatible with ASCII.

Unicode was designed to solve all of these problems. Simply put, it gives every individual human-designed character its own unique number, or “code point”. Doing this removes the ambiguity associated with displaying text. So long as a program supports Unicode, it doesn't need to know the language family that's being used. English, French, Arabic and Russian all can coexist on the same page, without any interference between the characters. Moreover, Unicode supports a very large number of code points, allowing Chinese and Japanese characters to coexist with alphabetic characters.

Encodings

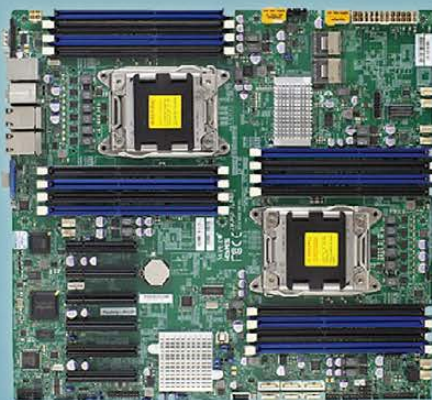
So far, so good. But, switching over to this new system raised two questions. First, how do you take these individual code points, uniquely identifying just about every character humans

have created, and translate them into bytes? Second, what happens to existing documents, which weren't written in Unicode?

On the one hand, the answers to those questions are relatively straightforward. On the other hand, the answers lead to much of the frustration associated with using Unicode—not because Unicode itself is bad or difficult, but because the mix of different, existing encodings with a Unicode-based system can be frustrating.

The first question, how do you encode the various Unicode characters using bytes, has multiple answers. If you're using a Unicode-aware language, you no longer can think of characters as being equivalent to bytes. Rather, one character might be a single byte, but it also might be multiple bytes. In the UCS-32 encoding scheme, for example, each Unicode character uses 4 bytes. This provides enough space for all of the defined Unicode characters, which is a good thing, but it also breaks

Petabyte Storage for 90K



(Rear View)



backward compatibility with ASCII documents and quadruples the size of anything written using ASCII or any of the ISO-8859 series.

For these reasons, the de facto standard in the Unicode world is UTF-8, a variable-length encoding scheme invented by famed programmers Rob Pike and Ken Thompson. The basic idea is that all defined ASCII characters, from 0–127, remain as they were. If the high (8th) bit is set, that indicates the character consumes an additional byte (that

is, two bytes for the character). In a similar way, high bits are used on succeeding bytes to indicate that the character's description has not ended. In this way, UTF-8 characters can consume as little as one byte (for ASCII characters) or as many as 6 bytes for truly unusual characters. Languages like Chinese and Japanese will require 4 bytes per character.

UTF-8 provides the best of all possible worlds—ASCII documents remain as they were, alphabetic languages don't use too many more

(1152 TB of Total Storage)

4x 4U Storage Nodes, 16 rack units, 288 Drives total

King Star 4U 72 x HDD Storage Server per Node

Dual Intel Xeon E5-2620 Six-Core 2.0GHz (2.50GHz Turbo)

-Intel® i350 GbE LAN (4 ports)

-Integrated IPMI 2.0 and KVM Over LAN

-Matrox G200eW VGA

-Triple 1280W(2+1) Redundant High-efficiency Digital Power Supplies

32GB DDR3 1600 ECC/REG Memory

72x 4TB SATA III Drives (Enterprise upgrade Available)

2 x 250GB SATA 6.0Gb/s 7.2K RPM (OS Drives, Raid 1)

SERVICE 3 Year Cross-Ship Service & Warranty

(Part# KSCT5237)

KSC

KING STAR COMPUTER

Rackmount Server Specialist

bytes than necessary, you resolve ambiguity with Unicode, and you can represent all Unicode characters. But, it does introduce a new problem: strings can now be invalid! If you were to use the fixed-width UCS-32 system, just about every byte would point to a valid character. But in UTF-8, it's possible to have a sequence of bytes that's invalid according to this encoding scheme.

To return to my example from earlier in this article, let's say I execute the following code in Python 3, rather than Python 2.7:

```
s = open('Downloads/test.jpg').read()
```

Now, in Python 2.7, strings are just collections of bytes. If I want to use Unicode, I need to use a "Unicode string", a special version of the `str` type in which characters are all in Unicode (and stored in UTF-8). In Python 3, the default string encoding is UTF-8, which means that executing the above code actually will result in an exception:

```
UnicodeDecodeError: 'utf-8' codec can't decode byte 0xff in
position 0: invalid start byte
```

In other words, Python was expecting to get input in UTF-8, but noticed the byte `0xFF` at the start of

the file, which is illegal. What you need to do is tell Python that you want to read the file in binary format, by opening it in "read binary" mode:

```
s = open('/Users/reuven/Downloads/test.jpg', mode='rb').read()
```

Now, given that you've read the file in binary mode, you're treating it as bytes, rather than a string. And sure enough, if you ask Python what type of data was returned:

```
>>> type(s)
<class 'bytes'>
```

In other words, Python won't create an illegal string. So instead of doing so, `read()` returns a bytestring, which is roughly the same as the Python 2.x string.

That covers files that were written in Unicode. But what about files written in another encoding scheme, such as ISO-8859-5? In such a case, you need to pass another parameter to "open", indicating the encoding you should use.

Ruby has undergone a similar change in the past few years. Ruby 1.8 saw strings as collections of bytes, but it really didn't think or care much about Unicode and other encodings. Ruby 1.9 (as well as 2.0) made a shift in a similar direction to Python, such

that every string has an encoding associated with it. Unlike Python, you can read binary data into a Ruby 2.0 string, and the language will be fine with that:

```
s = File.read('Downloads/test.jpg')
```

If you ask Ruby what sort of object was returned, it'll tell you that it was a string:

```
>> s.class #=> String
>> s.encoding #=> #<Encoding:UTF-8>
>> s.valid_encoding? #=> false
```

But, you then can set the encoding to something else:

```
>> s.force_encoding(Encoding.find('ASCII-8BIT'))
>> s.encoding #=> #<Encoding:ASCII-8BIT>
>> s.valid_encoding? #=> true
```

Web Development and Unicode

All of this is well and good, but how does it affect Web developers? Again, none of this would be a problem if you magically could flick a switch and have all documents and computers switch to using UTF-8. But, that's far from the case. Not only are there many documents out there that were written in non-UTF-8 formats, but also there are many computers whose

encoding is still not UTF-8.

This means if you have an HTML form and you accept input from users' browsers, you likely will get input from users' browsers in whatever encoding system their computers are using. True, most modern computers and browsers use UTF-8, but you would be amazed by how many old systems exist. You should experiment with your Web application, ensuring that even when someone sends you data in a non-Unicode system, you still can handle it (or gracefully deal with the failure).

Another issue I recently encountered myself wasn't directly from user input, but rather files that users were uploading. My Web application worked in UTF-8, and everything seemed to be humming along—until it wasn't. The problem was that part of the application involved people uploading text files. I would read the contents of the file into a string and then store that string in a database. Unfortunately, the application would raise an exception, because the text files—coming from people around the world, in different languages and using many different encodings—often were incompatible with UTF-8. One solution would have

been to try to identify the encoding of the uploaded file. In my particular case, I was able to catch the exception and report it to the user, indicating that only files in UTF-8 were acceptable. Whether such an error message will suffice for your application depends on what you're doing.

And yes, that leads me to my next point, namely databases. All of the major relational and NoSQL databases with which I work support UTF-8 as a default. PostgreSQL, for example, gives each database an encoding, indicating the encoding that will be used in text columns. The good news is that this ensures that all text stored in the database will be valid UTF-8, or whatever other encoding you use. The bad news (to some degree) is that if you want to store both binary and textual data in the same column, you'll have to find another solution. Binary data, such as the contents of a JPEG file, cannot be stored in a text column, because it's not legal UTF-8. Instead, you'll need to store such information in a binary BYTEA column, which accepts any sequence of bytes and doesn't attempt to ensure its validity. Fortunately, the drivers with which I work understand the difference

between TEXT and BYTEA columns and return results using appropriate data types.

Realize that there is a difference, however, between encoding and collation. *Encoding* refers to the way UTF-8 (or any other character set) is translated into a series of bytes. *Collation* refers to how the text is sorted and, thus, is language-dependent. Consider that sorting a list of 100 words will have different results in English, Spanish and French, and you'll understand that your application's needs (and users) will determine, to a large degree, which collation, if any, you choose to use.

Conclusion

Just about ten years ago, I worked on a multilingual site that required Unicode, and my decision to use it caused a great deal of friction with others working on the project, because they didn't have editors that supported UTF-8.

Things are quite different today. Just about every piece of Web-related software supports Unicode, from the operating system and language to the database and browser. However, the numerous non-Unicode computers, programs and files out there require that you



DAVE TAYLOR

Cribbage: Pairs and Three of a Kinds

Debugging his last article's script and calculating straights in a *Cribbage* hand keep Dave busy coding this month, with punctuation graffiti included!

The *Cribbage* game programming continues with further expansion of the subhand evaluation code. You'll recall that in a two-player game of *Cribbage*, you're dealt six cards but have to put two into the "crib", a third hand that alternates between players.

The challenge is this: which four cards of the six leave you with the most points possible?

There's a secondary consideration, because you also want to avoid putting points in the crib when it's not yours, if you can help it, but for now, I'm going to stick with the six-choose-four challenge.

And a challenge it is, because cards are worth points based on whether they have the same rank (for example, 9S and 9C = 2 points for a pair); whether they add up to 15, with all face cards = 10 (for example, 7S and

8C = 15 = 2 points); whether all four cards have the same suit (for example, 3D, 7D, 9D, QD = 4 points); and whether three or four of the cards are in sequential rank order (for example, 9D, 10C and JS = 3 points), even if they aren't the same suit.

I wrapped up my last article with code that could figure out the six-choose-four combinations (it's a straight combinatorics problem—I knew that stuff I learned in college eventually would come in handy), then evaluate each four-card set for possible fifteens and pairs. With some debugging code added, the current output looks like this:

```
$ sh cribbage.sh
Hand: 2C, 4S, 6S, 8C, 8H, 9C.
Subhand 0: 2C 4S 6S 8C
calc15() given ranks: 2 4 6 8
```

```

total 15-point value of that hand: 0

Subhand 1: 2C 4S 6S 8H
calc15() given ranks: 2 4 6 8
total 15-point value of that hand: 0

Subhand 2: 2C 4S 6S 9C
calc15() given ranks: 2 4 6 9
total 15-point value of that hand: 4
...

```

As you can see, the third subhand is worth more than the first two. In fact, 2C + 4S + 9C and 6S + 9C are both fifteens, so it's worth four points. Not too bad.

Further down in the debugging output, subhands start to appear with the pair of eights:

```

Subhand 6: 2C 6S 8C 8H
calc15() given ranks: 2 6 8 8
we've got a pair: 8 and 8
total 15-point value of that hand: 0

```

So at this point the code recognizes pairs, but the point accumulator isn't actually scoring them. That's not good.

Let's start by fixing that. The scoring code is getting pretty long, so I'll just share the two-card code, which is a bit simpler too:

```

for subhand in {0..5}
do

```

```

sum=0

for thecard in ${fourtwo[$subhand]}
do
    sum=$(( $sum + ${c15[$thecard]} ))
done

if [ $sum -eq 15 ] ; then
    points=$(( $points + 2 ))
fi

# now let's look at pairs
# remember: ${string:position:length}

twocards=${fourtwo[$subhand]}
card1=${twocards:0:1}
card2=${twocards:2}

if [ ${cr15[$card1]} = ${cr15[$card2]} ] ; then
    echo "we've got a pair: ${cr15[$card1]} and
        ${cr15[$card2]}"
    points=$(( $points + 2 ))
fi

done

```

Here's the line that fixed the scoring problem for pairs:

```
points=$(( $points + 2 ))
```

It's easy shell math, and something I hope you're using with some frequency. In fact, `$ ()` for subshells and `$ (())` for math equations that alternatively could be handled by `eval` are good.

That single line fixes the problem, as demonstrated in the very first test run:

```
Hand: 3H, 3D, 4C, 8H, 9H, JH.
Subhand 0: 3H 3D 4C 8H
calc15() given ranks: 3 3 4 8
we've got a pair: 3 and 3
total point value of that hand: 6
```

How did I get six points? 3H + 3D is a pair (2 points), then 3H + 4C + 8H = 15 (2 points) and 3D + 4C + 8H = 15 (2 points). That's a pretty decent little four-card *Cribbage* hand, actually.

What about when there are three cards that are the same? It turns out that *Cribbage* has a very logical scoring system, and three of a kind are scored as 3 * 2-card pairs, which makes sense. Here's an example to illustrate:

```
Hand: 4C, 4D, 4H, 7D, 10H, JS.
Subhand 0: 4C 4D 4H 7D
calc15() given ranks: 4 4 4 7
we've got a pair: 4 and 4
we've got a pair: 4 and 4
we've got a pair: 4 and 4
total point value of that hand: 12
```

So 4C + 4D, 4C + 4H and 4D + 4H are the three pair and are worth six points. This subhand is really superb, however, because there also are a number of card combinations that add up to fifteen, totaling 12 points. Very

good for four cards!

The piece that's missing with the scoring is straights. This is going to get a bit complicated, so stick with me.

Calculating Straight Runs

There's already code in place that generates all three-card combinations that catches when three cards sum up to fifteen points, so that's easily tapped within a "for" loop to extract the three-card index values:

```
combo=${fourthree[$subhand]}
```

That's going to be set to "0 1 2", "0 1 3" and so on.

The card's normalized rank (for example, J=10, Q=10) is set in the point calculation function as the local array `$cardrank[]`, and the original rank (J=11, Q=12 and so on) is in `$cardrankfull[]`. These originally were `c15[]` and `cr15[]`, but I renamed them to make their purpose a bit clearer in the script.

With "combo" set to the card indices, the full rank of a specific card in the four-card subhand can be referenced like this:

```
${cardrankfull[${combo:0:1}]}
```

As Douglas Adams would say, don't panic. Let's unwrap it instead.



KYLE RANKIN

Two Pi R 2: Web Servers

Who knew that Raspberry Pis would make such a great redundant Web cluster?

In my last column, I talked about how even though an individual Raspberry Pi is not that redundant, two Pis are. I described how to set up two Raspberry Pis as a fault-tolerant file server using the GlusterFS clustered filesystem. Well, now that we have redundant, fault-tolerant storage shared across two Raspberry Pis, we can use that as a foundation to build other fault-tolerant services. In this article, I describe how to set up a simple Web server cluster on top of the Raspberry Pi foundation we already have.

Just in case you didn't catch the first column, I'll go over the setup from last month. I have two Raspberry Pis: Pi1 and Pi2. Pi1 has an IP address of 192.168.0.121, and Pi2 has 192.168.0.122. I've set them up as a GlusterFS cluster, and they are sharing a volume named gv0 between them. I also mounted this shared volume on both machines

at /mnt/gluster1, so they each could access the shared storage at the same time. Finally, I performed some failure testing. I mounted this shared storage on a third machine and launched a simple script that wrote the date to a file on the shared storage. Then, I experimented with taking down each Raspberry Pi individually to confirm the storage stayed up.

Now that I have the storage up and tested, I'd like to set up these Raspberry Pis as a fault-tolerant Web cluster. Granted, Raspberry Pis don't have a speedy processor or a lot of RAM, but they still have more than enough resources to act as a Web server for static files. Although the example I'm going to give is very simplistic, that's intentional—the idea is that once you have validated that a simple static site can be hosted on redundant Raspberry Pis, you can expand that with some more sophisticated content yourself.

Install Nginx

Although I like Apache just fine, for a limited-resource Web server serving static files, something like nginx has the right blend of features, speed and low resource consumption that make it ideal for this site. Nginx is available in the default Raspbian package repository, so I log in to the first Raspberry Pi in the cluster and run:

```
$ sudo apt-get update
$ sudo apt-get install nginx
```

Once nginx installed, I created a new basic nginx configuration at `/mnt/gluster1/cluster` that contains the following config:

```
server {
    root /mnt/gluster1/www;
    index index.html index.htm;
    server_name twopir twopir.example.com;

    location / {
        try_files $uri $uri/ /index.html;
    }
}
```

Note: I decided to name the service `twopir`, but you would change this to whatever hostname you want to use for the site. Also notice that I set the document root to `/mnt/gluster1/www`. This way, I

can put all of my static files onto shared storage so they are available from either host.

Now that I have an nginx config, I need to move the default nginx config out of the way and set up this config to be the default. Under Debian, nginx organizes its files a lot like Apache with `sites-available` and `sites-enabled` directories. Virtual host configs are stored in `sites-available`, and `sites-enabled` contains symlinks to those configs that you want to enable. Here are the steps I performed on the first Raspberry Pi:

```
$ cd /etc/nginx/sites-available
$ sudo ln -s /mnt/gluster1/cluster .
$ cd /etc/nginx/sites-enabled
$ sudo rm default
$ sudo ln -s /etc/nginx/sites-available/cluster .
```

Now I have a configuration in place but no document root to serve. The next step is to create a `/mnt/gluster1/www` directory and copy over the default nginx `index.html` file to it. Of course, you probably would want to create your own custom `index.html` file here instead, but copying a file is a good start:

```
$ sudo mkdir /mnt/gluster1/www
$ cp /usr/share/nginx/www/index.html /mnt/gluster1/www
```

With the document root in place, I can restart the nginx service:

```
$ sudo /etc/init.d/nginx restart
```

Now I can go to my DNS server and make sure I have an A record for twopir that points to my first Raspberry Pi at 192.168.0.121. In your case, of course, you would update your DNS server with your hostname and IP. Now I would open up `http://twopir/` in a browser and confirm that I see the default nginx page. If I look at the `/var/log/nginx/access.log` file, I should see evidence that I hit the page.

Once I've validated that the Web server works on the first Raspberry Pi, it's time to duplicate some of the work on the second Raspberry Pi. Because I'm storing configurations on the shared GlusterFS storage, really all I need to do is install nginx, create the proper symlinks to enable my custom nginx config and restart nginx:

```
$ sudo apt-get update
$ sudo apt-get install nginx
$ cd /etc/nginx/sites-available
$ sudo ln -s /mnt/gluster1/cluster .
$ cd /etc/nginx/sites-enabled
$ sudo rm default
$ sudo ln -s /etc/nginx/sites-available/cluster .
$ sudo /etc/init.d/nginx restart
```

Two DNS A Records

So, now I have two Web hosts that can host the same content, but the next step in this process is an important part of what makes this setup redundant. Although you definitely could set up a service like heartbeat with some sort of floating IP address that changed from one Raspberry Pi to the next depending on what was up, an even better approach is to use two DNS A records for the same hostname that point to each of the Raspberry Pi IPs. Some people refer to this as DNS load balancing, because by default, DNS lookups for a hostname that has multiple A records will return the results in random order each time you make the request:

```
$ dig twopir.example.com A +short
192.168.0.121
192.168.0.122
$ dig twopir.example.com A +short
192.168.0.122
192.168.0.121
```

Because the results are returned in random order, clients should get sent evenly between the different hosts, and in effect, multiple A records do result in a form of load balancing. What interests me about a host

What is a **Superhero** without the right tools?



No, we're not referring to costumed crime fighters who come to the rescue of others in trouble. We're talking about IT operations personnel - people with complex skills and a daunting job, who fight downtime, performance slowdowns, and other evils to keep your apps running 24x7. They have to be ready to take action if there is trouble brewing in the system – and being ready involves having the right tools.

ManageEngine provides the right set of monitoring tools for your IT operations team, enabling them to keep track of the performance of their complex apps from both within and outside their data center.

www.manageengine.com/apm

www.site24x7.com

- ⊙ Application Performance Monitoring
- ⊙ End User Experience Monitoring
- ⊙ Anomaly Detection
- ⊙ Automated Dependency Mapping
- ⊙ Deep Transaction Monitoring
- ⊙ Integrated Management Console



Register Now!

2013 USENIX Federated Conferences Week

June 24–28, 2013 • San Jose, CA

www.usenix.org/conference/fcw13

USENIX ATC '13

2013 USENIX Annual
Technical Conference
Wednesday–Friday, June 26–28
www.usenix.org/atc13

ICAC '13

10th International Conference on
Autonomic Computing
Wednesday–Friday, June 26–28
www.usenix.org/icac13

HotPar '13

5th USENIX Workshop on
Hot Topics in Parallelism
Monday–Tuesday, June 24–25
www.usenix.org/hotpar13

UCMS '13

2013 USENIX Configuration
Management Summit
Monday, June 24
www.usenix.org/ucms13

Feedback Computing '13

8th International Workshop on
Feedback Computing
Tuesday, June 25
www.usenix.org/feedback13

ESOS '13

2013 Workshop on
Embedded Self-Organizing Systems
Tuesday, June 25
www.usenix.org/esos13

HotCloud '13

5th USENIX Workshop on
Hot Topics in Cloud Computing
Tuesday–Wednesday, June 25–26
www.usenix.org/hotcloud13

WiAC '13

2013 Women in Advanced
Computing Summit
Wednesday–Thursday, June 26–27
www.usenix.org/wiac13

HotStorage '13

5th USENIX Workshop on
Hot Topics in Storage and
File Systems
Thursday–Friday, June 27–28
www.usenix.org/hotstorage13

HotSWUp '13

5th Workshop on Hot Topics
in Software Upgrades
Friday, June 28
www.usenix.org/hotswup13

Registration Discounts Available!

Register by the Early Bird Deadline,
Monday, June 3, and save!

AND MORE!

Stay Connected...



www.twitter.com/usenix



www.usenix.org/youtube



www.usenix.org/facebook



www.usenix.org/linkedin



www.usenix.org/gplus



www.usenix.org/blog





SHAWN POWERS

Prospecting for Ones and Zeros

Create money out of thin air with cryptocurrency.

At the current market rates (April 24, 2013), a user with a single AMD Radeon 7950HD video card can make \$6 a day mining cryptocurrency. Granted, it'll cost around \$1 in electricity, but that's still a \$5-per-day profit. Of course, tomorrow the market could tank, and it'll be worthless, but at this precise moment, it's profitable. I've written about cryptocurrency before in *Linux Journal*, but as you can imagine, I've gotten lots of questions about mining since the price skyrocketed. In this article, I want to talk specifically about mining. If you're unsure about cryptocurrency in general, look back at my past article: <http://www.linuxjournal.com/content/cryptocurrency-your-total-cost-01001010010>. If you just want to get down and nerdy, keep reading.

CPUs or GPUs?

Cryptocurrencies come in several varieties, but the most popular, and most valuable, is the Bitcoin. Since its early days, it's been silly to mine Bitcoins with a CPU. The raw horsepower of a GPU blows the doors off any CPU, regardless of the CPU's speed. Because of that wasted opportunity, Litecoin was developed specifically to make mining difficult for GPUs.

For several months, it proved to be a great way to take advantage of idle CPUs. It also opened up mining to the regular desktop user, because a standard workstation CPU would be able to contribute and create coinage. Brilliant programmers being what they are, however, GPUs now are able to mine Litecoins more efficiently than CPUs. It's still profitable to mine Litecoins with high-end CPUs, but

Although Bitcoins still are the most popular cryptocurrency, an interesting mining fact is that it's more profitable to mine Litecoins and then trade them for Bitcoins.

the profits are measured in pennies instead of dollars.

Although Bitcoins still are the most popular cryptocurrency, an interesting mining fact is that it's more profitable to mine Litecoins and then trade them for Bitcoins. Perhaps that's my own little trade secret, but it's held true for several months, and it looks to continue. Once you figure out the hash rate you can achieve with Bitcoins versus your Litecoin hash rate, it's as simple as plugging the numbers in to the calculator at <http://allchains.info/calc> to see which is more profitable.

Where to Begin?

If you want to get into mining, it certainly sounds like a good idea to start with CPU mining on your existing desktop computer. Although that's definitely possible, my fear is that it will cause more frustration than fun. The profits, if any, will be in the pennies. If the market fluctuates (which it *always* does), you could go from making three cents a day to

losing a nickel a day overnight. Plus, your system will be running at 100% CPU usage, making the rest of your computing frustrating. Still, if you just want to dabble, it's a way to start. If you want to mine with your desktop machine, you'll need to use a mining pool; otherwise, it will take years before you see any profit.

To start with serious mining, it means purchasing video cards. For Bitcoins, there also are custom FPGA and ASIC boards that can do mining, but most people still use GPUs, so I focus on those here. Also, in this article, I focus more on Litecoins because they're slightly more profitable to mine, but also because the mining rigs are harder to configure. If you can mine Litecoins, Bitcoins are a breeze, so let's focus on Litecoins.

Currently, the best price/hashrate/electrical-usage video card is the AMD 7950HD. For about \$300, the 7950 produces around 600kH/s on the Litecoin network. There is a great wiki that compares the various GPU

chipsets and their wattage versus hashrate. You owe it to yourself to study the wiki to see what will work best for you: <https://github.com/litecoin-project/litecoin/wiki/Mining-hardware-comparison>.

(There is a similar chart for Bitcoin mining. It's worth reading that in case the Litecoin market crashes and burns, and being able to switch to mining Bitcoins efficiently is a bonus: https://en.bitcoin.it/wiki/Mining_hardware_comparison.)

In addition to the hash rate and electrical usage, the allchains.info site mentioned previously is a great tool for figuring out how much money a particular set of GPUs will create. Keep in mind that the market is highly volatile, but it's a nice way to see a

snapshot daily rate.

And of course, to use a GPU, you need a motherboard that will hold it. Thankfully, cryptocurrencies don't require anything more than PCI-E 1x for bandwidth, so as long as you can get the cards into the slots (or use extenders), you're set. Add a cheap CPU, a gig or two of RAM and a power supply big enough to support your rig, and you're golden! I know, it's a lot to think about and a lot to consider, but if this sort of planning is fun, mining cryptocurrency is perfect for you!

My Head Hurts

Did the last section dash your hopes of ever becoming a miner? Too complicated? Too many variables? I



Figure 1. The PCI-E slots are far enough apart that all three are usable.

understand. Here's a sample mining rig, part by part:

Motherboard: ASRock 970 EXTREME4—this motherboard is about \$99 and has the convenient spacing that allows (three) GPUs to be mounted at the same time (Figure 1). They'll be crammed together, but I'll talk about cooling a little later.

RAM: Crucial 2GB Kit (2x1GB) 240 pin DDR3 1333—for \$30 or so, this is more than enough RAM to mine with. When I tried to find a 1GB kit, it cost more, so I got this kit because it was the cheapest.

CPU: AMD Sempron 145 Sargas 2.8GHz—this single-core CPU is less than \$40, fits in the AM3+ slot on the motherboard listed above and is horrible for mining anything. You could buy an expensive CPU and get some mining hashes out of it, but the time it would take to recoup the expense is crazy. I recommend slow and cheap.

Hard Drive: anything—if you're considering building a mining rig, chances are you have a pile of old hard drives sitting in a closet or on a shelf. Just use one of those. Otherwise, any old SATA hard drive will work.

Power Supply: 850Watt 80 PLUS certified or better—don't go less than 850 watts, even if you don't start with

three full cards. Give yourself room to grow. Also, the more efficient power supply you buy, the more profitable your rig will be in the long run. The price you pay will vary anywhere from \$90 to \$175 or so. Make sure you have enough PCI-E power connectors for the cards you're going to be installing. (Note: you can get adapters to convert SATA power connectors to PCI-E connectors, but I wouldn't recommend using too many of those in a single rig.)

GPUs: AMD Radeon 7950 HD—these seem to be available from a fairly wide variety of on-line sources. The prices range from \$299 to around \$329, and each card will do around 600kH/s on the Litecoin network. Other cards work, some are faster, and if you can get a deal on something else, just compare on the charts mentioned previously. If you just want a parts list, however, go with the 7950. It's a great card.

Case: don't use a case. GPUs produce a *ton* of heat while they're mining, so it's best to have them out in the open air. Plus, you can get a \$10 box fan and push more cooling air over an exposed rig than any number of case fans could do.

The Build

If you've researched for days and

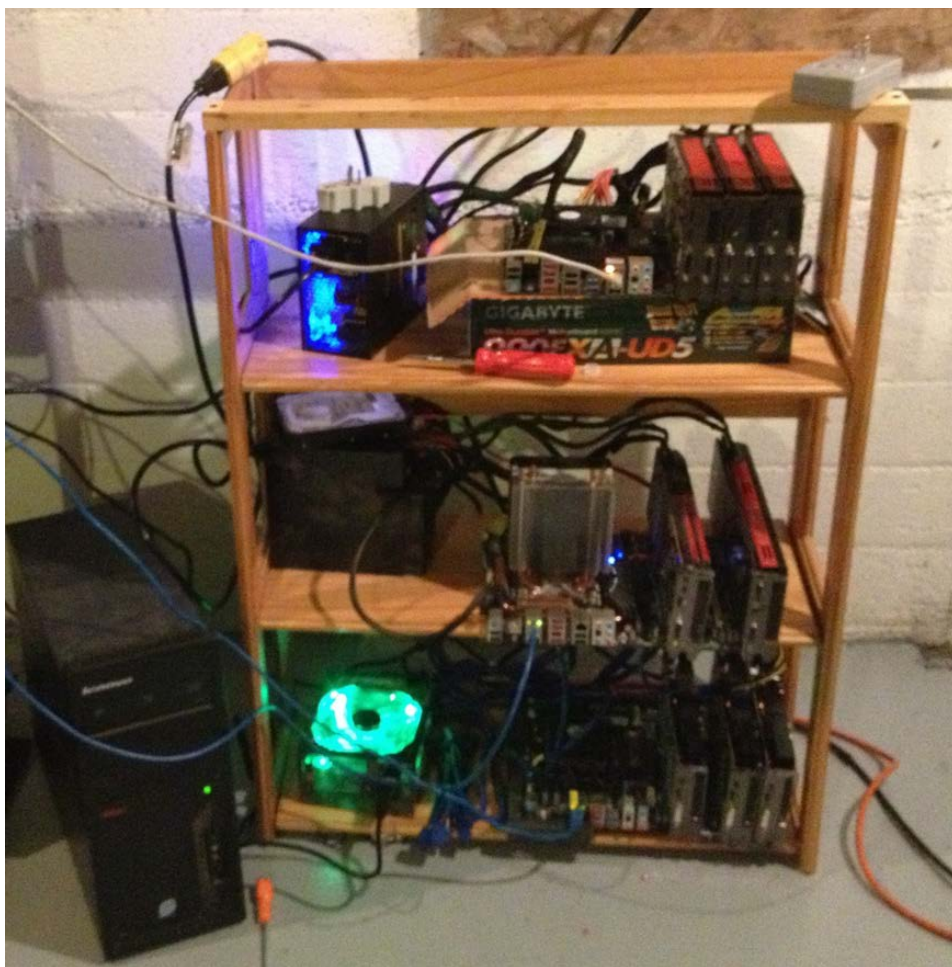


Figure 2. It's not pretty, but it does the job!

ordered what you found to be the absolute best possible combination of hardware, or if you just used my list, the build is similar. When not using a case, find a fairly secure location where you can set up computer equipment and not worry about anyone bumping it. (I have my rigs set up in the basement; see Figure 2.)

I should add a disclaimer that you should wear a static wrist strap, make sure everything is grounded,

and securely mount the motherboards and power supplies onto whatever surface you're using. Those are all very important things. As you can see in my photo, however, my motherboards are sitting on empty boxes, and I turn my rigs on by shorting the power connector on the motherboard with a screwdriver. My setup is a bit redneck, and I can't recommend that you do the same,

but I won't judge you if you do.

These rigs will be headless, as you can tell by the photo, but for the operating system install, you'll need a monitor. I recommend installing Xubuntu. I won't go through the details on how to install Xubuntu, but you can use a USB stick to install it or a temporary CD drive—whatever floats your boat. *Be sure* to have the system log in automatically. This is a requirement for later on when you won't have a monitor connected!

Unless you have many thousands of dollars worth of mining rigs, you'll want to mine at a pool.

Once Xubuntu is installed, you need to go to <http://amd.com> and get the latest Catalyst driver for Linux. (Make sure to get the version that matches your installed system in regards to 32- or 64-bit.) Please note that although there are proprietary drivers in the Xubuntu distro, those drivers do not have the things required for mining. You'll need to install the driver from amd.com in order to mine properly.

After you run the installer, you'll be forced to reboot. Once rebooted, you need to make sure the video card(s) are all initialized. This is a squirrely procedure, which doesn't work consistently. Simply open a terminal and type:

```
sudo aticonfig --adapter=all --initial
```

The squirrely part is that sometimes it doesn't detect all the cards. Look at the output, and make sure it listed the number of cards you have installed. If not, run the same command over, and it likely will detect them. (I know, it's weird.) Once that's done, you'll need to reboot again. At that

point, configuring the GPU drivers should be done.

The other tools you'll need installed (these from the product repositories) are screen and openssh-server. Those come into play when you run the miners in headless mode. Make sure you have a static IP address set on your mining rig, then reboot it one final time, disconnecting the keyboard, mouse and monitor. If everything goes right, you should have a headless miner set to log in automatically, with an SSH server running so you can connect remotely. Leave the dark basement and go back to your comfy workstation. You should be able to do the rest from there.

A Bit about Pools

Before you actually set up the mining programs, it's important to decide where you're going to mine. Back in the day before cryptocurrencies were popular, a user would install the litecoin client and mine locally. The problem with that method now is that it would take weeks or months to find a "block" mining on your own. Unless you have

many thousands of dollars worth of mining rigs, you'll want to mine at a pool. Basically, users at a pool will combine their mining horsepower and then split the coins they mine proportionally to the hashing power they contribute. This means instead of mining for months and possibly finding a block of 50 coins, the pool miners get a regular payout daily based on their contribution to the network. With my nine GPUs mining 24/7, I can earn only ten Litecoins a day. If I didn't mine with a pool, it would become frustrating quickly.

Several mining pools are available for Litecoins and Bitcoins. Regardless of what pool you choose, I recommend withdrawing your earnings on a regular basis, because pool owners are anonymous, and I'm not terribly trusting. (Perhaps *Linux Journal* should start a pool—hmmm....) Here's a partial list of pools: <https://github.com/litecoin-project/litecoin/wiki/Comparison-of-mining-pools>.

One Miner to Rule Them All

I know I'm focusing on GPU mining here, but since I'm talking about miner programs, it seems a good time at least to mention *how* to CPU mine. If you're mining Litecoins with a CPU, you want to use pooler's

miner (<https://github.com/pooler/cpuminer>). For mining with a GPU, whether you want to mine Bitcoins or Litecoins, cgminer is the tool of choice (<https://github.com/ckolivas/cgminer>). I use cgminer here and configure it to mine Litecoins. Be sure to get the latest binary, because Litecoin support is a fairly recent addition to cgminer.

From the comfort of your workstation, ssh in to your miner. Assuming everything is working correctly, you should be able to log right in. Now that you're logged in, type `screen` to get a screen session going, so once you start mining, you can disconnect and it will keep mining. Because mining requires OpenCL, it means that you have to be using the X Window System display. Because you're SSH'd in, that would be a problem, except that Linux is so very flexible. In the screen session, type:

```
export DISPLAY=:0
```

That should give you the ability to use OpenCL. While you're at it, export a couple other variables that fix a few oddities with how GPU memory is handled:

```
export GPU_USE_SYNC_OBJECTS=1
export GPU_MAX_ALLOC_PERCENT=100
```

There should be nothing returned from the system; you're just setting some environment variables so that cgminer works properly. If you haven't downloaded and extracted the cgminer program, do so now. Then, cd to that directory:

```
cd cgminer_folder_name
```

Now, see if cgminer sees all your GPUs:

```
./cgminer -n
```

You should see something like this:

```
[2013-04-24] CL Platform 0 vendor: Advanced Micro Devices, Inc.

[2013-04-24] CL Platform 0 name: AMD Accelerated Parallel Processing

[2013-04-24] CL Platform 0 version: OpenCL 1.2 AMD-APP (1084.4)

[2013-04-24] Platform 0 devices: 3

[2013-04-24] 0      Cypress
[2013-04-24] 1      Cypress
[2013-04-24] 2      Cypress

[2013-04-24] 3 GPU devices max detected
```

```
cgminer version 2.11.4 - Started: [2013-04-25 11:21:14]
-----
(5s):1.025M (avg):1.011Mh/s | A:83 R:0 HW:0 U:11.2/m WU:999.6/m
ST: 2 SS: 0 NB: 7 LW: 91 GF: 0 RF: 0
Connected to us.litecoinpool.org diff 128 with stratum as user trunkboy.1
Block: 2ac2517c50c7c097... Diff:26.5M Started: [11:28:15] Best share: 3.92K
-----
[P]ool management [G]PU management [S]ettings [D]isplay options [Q]uit
GPU 0: 72.5C 3470RPM | 339.0K/342.4Kh/s | A:20 R:0 HW:0 U: 2.69/m I:19
GPU 1: 73.5C 3131RPM | 338.9K/342.4Kh/s | A:33 R:0 HW:0 U: 4.44/m I:19
GPU 2: 73.0C 2141RPM | 339.0K/340.0Kh/s | A:30 R:0 HW:0 U: 4.04/m I:19
-----

[2013-04-25 11:27:10] Accepted 20833472 Diff 208/128 GPU 0 pool 0
[2013-04-25 11:27:22] Accepted 96e29d2f Diff 243/128 GPU 1 pool 0
[2013-04-25 11:27:24] Accepted c7bca281 Diff 185/128 GPU 0 pool 0
[2013-04-25 11:27:27] Accepted de1982b7 Diff 374/128 GPU 1 pool 0
[2013-04-25 11:27:30] Accepted 952bdc4b Diff 141/128 GPU 2 pool 0
[2013-04-25 11:27:31] Accepted eaac816c Diff 3.29K/128 GPU 1 pool 0
[2013-04-25 11:27:50] Accepted 64d9c847 Diff 317/128 GPU 1 pool 0
[2013-04-25 11:27:58] Stratum from pool 0 detected new block
[2013-04-25 11:28:02] Accepted 050c5969 Diff 332/128 GPU 1 pool 0
[2013-04-25 11:28:02] Accepted 7acecfea Diff 170/128 GPU 2 pool 0
[2013-04-25 11:28:07] Accepted 171a00b7 Diff 327/128 GPU 0 pool 0
[2013-04-25 11:28:08] Accepted 737bde00 Diff 131/128 GPU 1 pool 0
[2013-04-25 11:28:15] Stratum from pool 0 detected new block
```

Figure 3. These are actually three 5850 cards, not 7950s

Assuming all your cards are showing, you're ready to mine! If not, you might have to fiddle with the `aticonfig` tool again. Remember, it's a bit squirrely. Because you're inside a screen session, you can start mining and then disconnect the screen session without stopping `cgminer`. For a quick start, type something like this (all on one line):

```
./cgminer --scrypt -o stratum+tcp://your.pool:3333
-u pool_user -p pool_pass -I 13
```

You'll need to fill in your information for pool IP, port, user and password. If everything works correctly, after a few moments, you should see `cgminer` doing its thing with all your GPUs. (Figure 3 shows one of my rigs.)

Step 73—Profit?

To be honest, all this work got you only to the beginning of configuring your mining rigs. If there's enough interest, perhaps I'll do another column on tweaking and overclocking for maximum hash rates and maximum profit. In the meantime, reading forums like <https://bitcointalk.org> will yield some valuable information. (It also will yield trolls, scammers and so on—the forum is like the dirty underbelly of the Bitcoin

mining community.) The next steps I recommend are:

- Read the `cgminer` options and configuration options in the README. It's an amazing piece of software.
- Tweak your memory speeds, clock speeds, thread concurrency and intensities to get better hash rates.
- Set up scripts to start your miners, and create config files for `cgminer` instead of putting everything on a really long command-line argument. Use your script to export those environment variables so you don't forget to do so.
- Install the litecoin (or bitcoin) client on a computer, so you can

Resources

Bitcoin Forum: <https://bitcointalk.org>

Chain Data: <http://allchains.info>

Litecoin Project on GitHub:
<https://github.com/litecoin-project>

Litecoin: <http://litecoin.org>

Bitcoin: <http://bitcoin.org>

withdraw your coins to your own local wallet.

- Check out the Resources section for this article. Lots of information is available; it's a data-geek's dream.

If you're a cryptocurrency miner or are about to become one, please send photos of your mining rigs to ljeditor@linuxjournal.com. You can tell by my setup that we don't care about professionalism, it's just really cool to see other people's setups.

If you'd like to hear more about Bitcoins or Litecoins, let us know that too! ■

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

Send comments or feedback via <http://www.linuxjournal.com/contact> or to ljeditor@linuxjournal.com.

LINUX JOURNAL

now available
for the **iPad** and
iPhone at the
App Store.



linuxjournal.com/ios



For more information about advertising opportunities within *Linux Journal* iPhone, iPad and Android apps, contact John Grogan at +1-713-344-1956 x2 or ads@linuxjournal.com.

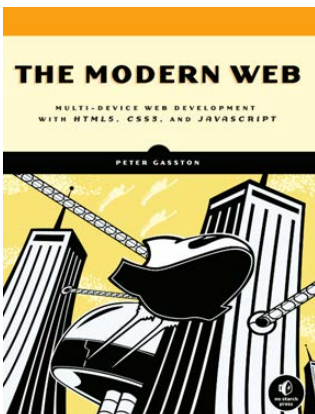
MSC Embedded Inc.'s Qseven Starter Kit

The new Qseven Starter Kit with AMD embedded G-Series APU, says its producer MSC Embedded Inc., frequently will be used to drive an LCD output. The MSC Q7-SK-A50M-EP4 Starter Kit consists of MSC's 3.5" Qseven baseboard Q7-MB-EP4 with heatspreader and heatsink and an integrated power supply with cable kit. The kit comes with a ready-to-run Linux installation in Flash Disk to enable an out-of-the-box functional experience.

An optional TFT kit is available, which provides for a 12.1"

LCD panel with XGA resolution (1024 x 768), the appropriate cable kit for operation off the Qseven baseboard and full implementation in the Qseven module's Graphics BIOS. Users can select the Qseven module with the most suitable processor and clock speed from the MSC range of Q7-A50M modules: AMD G-T40E Dual-Core APU or AMD G-T40R Single-Core APU with optional 7.2GB Flash Disk or AMD G-T16R Single-Core APU.

<http://www.msembedded.com>

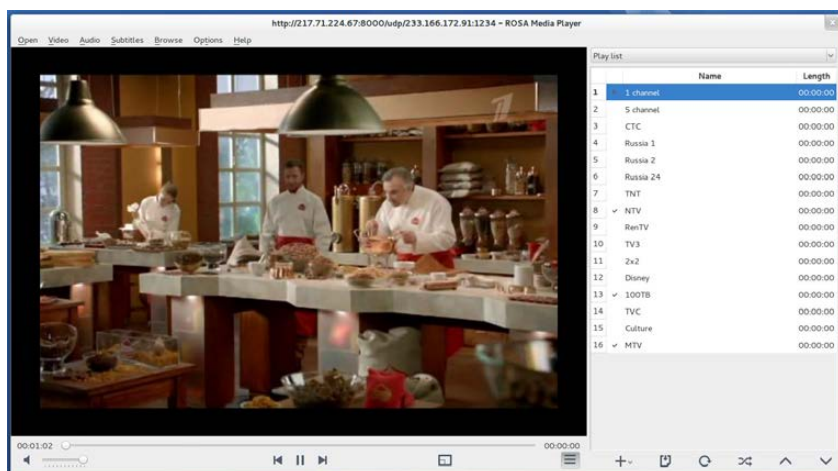


Peter Gasston's *Modern Web* (No Starch Press)

Today's Web technologies are evolving at near-light-speed. When users can browse the Web on a 3" phone screen as easily as on a 50" HDTV, what's a developer to do? Well, flummoxed friends, there is a new book designed to answer just that question, namely Peter Gasston's *Modern Web: Multi-Device Web Development with HTML5, CSS3, and JavaScript*. Gasston's book will guide readers

through the latest and most important tools for device-agnostic Web development, including HTML5, CSS3 and JavaScript. His plain-English explanations and practical examples emphasize the techniques, principles and practices one needs to transcend individual browser quirks easily and stay relevant as these technologies are updated. After reading the book, claims publisher No Starch Press, one will be equipped to design content that displays fluidly across multiple devices as well as turn outdated Web sites into flexible, user-friendly ones that take full advantage of the unique capabilities of any device or browser.

<http://www.nostarch.com>



ROSA Media Player

Handiwork of the Russian firm ROSA, the new ROSA Media Player (ROMP) 1.6 is reportedly the first free media solution with full YouTube support. This support enables users to search, view and save

YouTube video clips to their computers. Other advances in the 1.6 release include desktop video and audio capture, DVD menu support and improved IPTV support, as well as the ability to load a list of channels from a remote server. ROMP, available in Linux and Windows versions, is distributed under the GPL 3+ license and available for free download and via various distribution repositories.

<http://www.rosalab.ru>

Exablox's OneBlox

After two years in stealth development mode, Exablox has burst out of the lab to debut its OneBlox flagship solution, an affordable storage appliance that combines a paradigm-shifting architecture design with integrated, enterprise-grade software features. The two-year development phase, complete with extensive customer and partner research, chipped away at businesses' most common storage pain points, including runaway costs, complicated installation, cumbersome data management and a lack of data security. The 100% cloud-based OneBlox's key product highlights include initial installs in less than five minutes and subsequent configurations in less than three, HTML5 storage management and control that graphically depict storage use and performance, automatically enabled deduplication and compression, dynamic and automatic capacity and performance expansion with new drives or nodes, instant data recovery from any desktop and an encrypted local copy of all data.

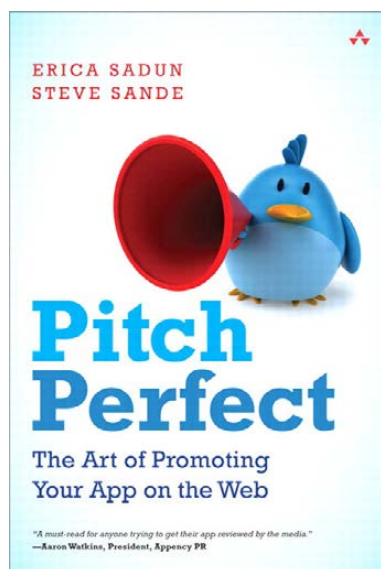
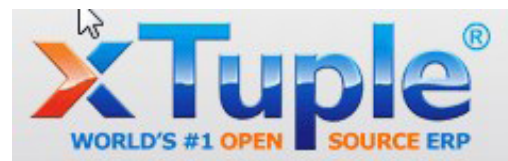
<http://www.exablox.com>



xTuple ERP

Distribution companies have three problems: items, items and items. Managing all the inventory in distribution-heavy industries, such as electrical, HVAC/R, PVF, office products and automotive, can be a management nightmare. These sectors are potential target customers for the updated xTuple ERP, which includes the new xWD extension for sophisticated, enterprise-class inventory control. One feature of xTuple's xWD is the external catalog that allows distribution companies to merchandise non-stock products readily for immediate sales fulfillment and generation of a corresponding purchase order. The xWD module complements the open-source xTuple's existing functionality in sales, accounting and operations—including customer and supplier management, inventory control, manufacturing and distribution. xTuple gives customers the ability to tailor solutions with multiplatform support for Linux, Windows, Mac and mobile.

<http://xTuple.org>



Erica Sadun and Steve Sande's *Pitch Perfect: The Art of Promoting Your App on the Web* (Addison-Wesley)

You're about to pour your heart, soul, skill...and savings, into creating the next big app. How will anyone know how awesome it is? Authors Erica Sadun and Steve Sande are veteran tech bloggers who have seen countless marketing successes and failures in the app world firsthand. Their new Addison-Wesley book *Pitch Perfect: The Art of Promoting Your App on the Web* is a guide to success with your

new app. The book provides an "in" with popular blogs to get those make-or-break product reviews and features extensive tutorials on how to develop short-and-sweet pitches that capture attention. It also offers tips on how to avoid Inbox deletion, how to keep tech bloggers in the loop during the development process and how to maintain good relations with the public and the press.

<http://www.informit.com>



6WIND's 6WINDGate

Readers interested in virtualization and enterprise networking solutions will want to read on about improvements to the 6WINDGate networking software from 6WIND. 6WINDGate solves critical performance and scalability challenges for virtual switches while retaining full compatibility with standard virtual switch software, such as Open vSwitch. The company claims that 6WINDGate provides an optimized data plane solution that delivers more than a 10x acceleration for the baseline switching functions and delivers high performance for secure tunneling protocols, such as IPsec, GRE, NVGRE, VLAN and VxLAN. This enables service providers, such as cloud and telecom data centers, to achieve significant CAPEX and OPEX 1 improvements that are not possible without data plane acceleration. The virtual switch acceleration delivered by 6WINDGate is transparent to the applications running on the platform, which do not need to be recompiled or re-verified in order to work with this high-performance solution.

<http://www.6wind.com>

Wind River Linux Carrier-Grade Profile

Although carrier-grade functionality isn't new for Wind River, the new Wind River Linux Carrier-Grade Profile for the latest version of Wind River Linux delivers these capabilities on top of a Yocto Project-compatible product. Formally registered for the CGL 5.0 specification with the Linux Foundation, the profile is the first delivery of carrier-grade Linux functionalities on top of a Yocto compatibility, says the company. The Yocto Project is a multivendor, open-source project that provides templates, tools and methods for creating custom Linux-based systems for embedded products, regardless of the hardware architecture. The net result of this combination of features, says Wind River, is a decrease in complexity and costs, an increase in the portability of Linux implementations and improved cross-platform compatibility and component interoperability. Carrier-grade products typically require up to 5 nines or 6 nines (99.999% to 99.9999%) availability, translating to downtime as low as 30 seconds a year.

<http://www.windriver.com>



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

AIDE

DEVELOPING FOR ANDROID ON ANDROID

No matter where you find yourself, there you are.
And with AIDE, now you can develop code there too.

JOEY BERNARD



Figure 1. AIDE is available as a free download from the Google Play store.

Android, as a platform, is one of the fastest growing on the planet. It is available on smartphones and a series of different tablet sizes. Most devices also include a full spectrum of sensors that are available to programs you install, so it's a very inviting platform for development. The usual workflow involves installing a development environment on some other machine, either a Windows or Linux desktop or laptop. You then do all of your code writing, compiling and debugging there before you actually copy it and install it onto your Android device.

But, there may be times when

you want to develop on the road or shorten the cycle by developing on your Android device itself. One of the better options for this is AIDE, the Android Java IDE. AIDE is distributed under a freemium model. The free version allows you to develop, compile and run your code. It also allows you to install to the device on which AIDE is running. However, if you want to generate APK files that can be used to install onto other devices, you need to purchase the full version.

In this article, I start by explaining how to install AIDE and create a new program, and then I cover what

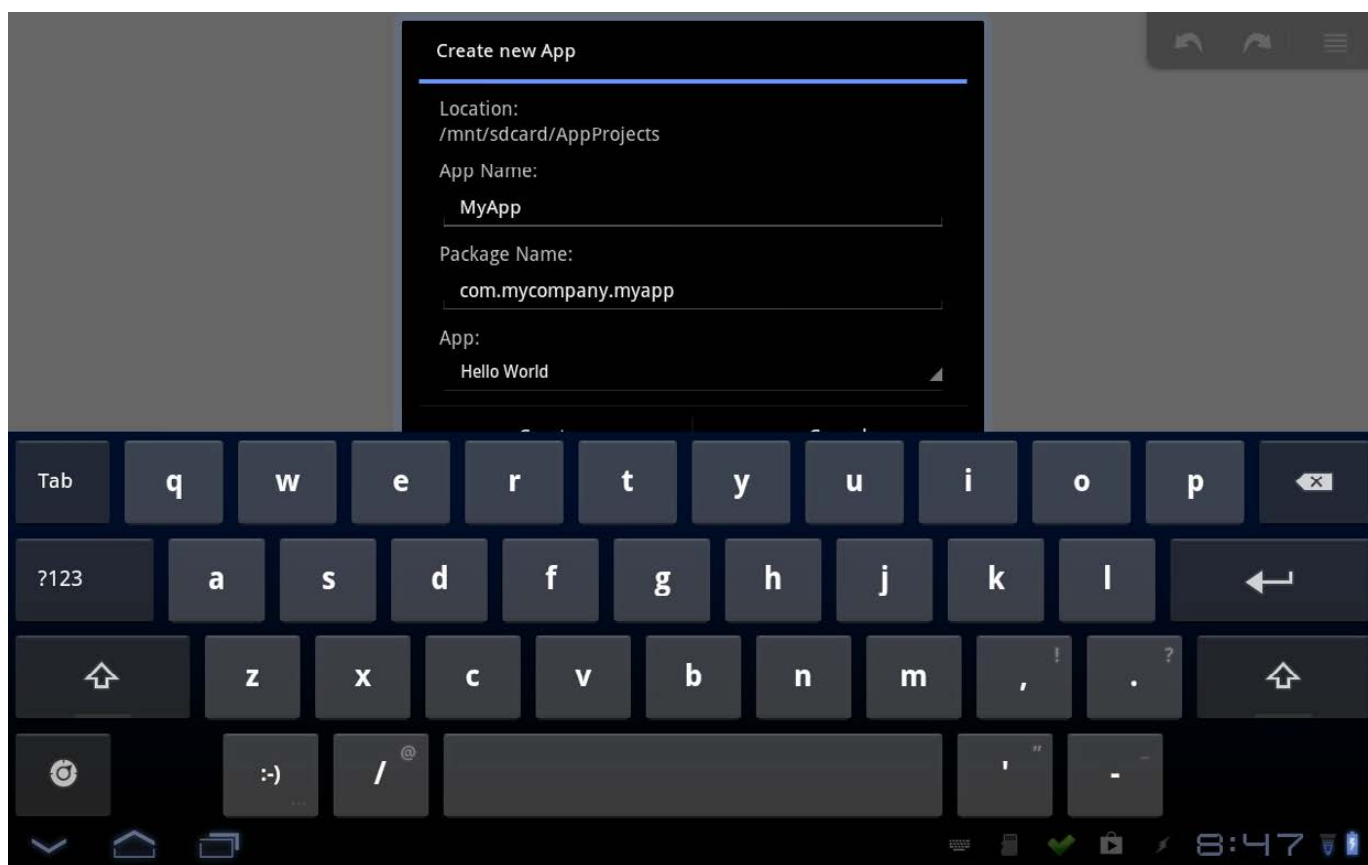


Figure 2. When AIDE starts up the first time, you are shown a dialog for your first project.

is involved in coding, debugging and running your new program. For more information, see the Google+ page (<https://plus.google.com/101304250883271700981/about>).

The first step is to install AIDE on your Android device. Open up the Play Store and do a search for “AIDE”. The appropriate package should show up at the top of the list. If you are in doubt, verify that the developer is “appfour GmbH”. AIDE takes up more than 12MB, so if you are running short on space, you can transfer the majority of it to

an SD card, leaving 4.45MB in your device’s main storage.

The very first time you start AIDE, it will pop up a dialog box where you can enter the details for beginning a project. In this dialog, you can enter an App Name and a Package Name. You also can select an app template from some built-in ones, such as “Hello World”, “Tetris” and “Analog Clock Widget”. These templates will set up the folders and files for your new project.

Once you click create, the files will be created in the folder `/mnt/sdcard/`

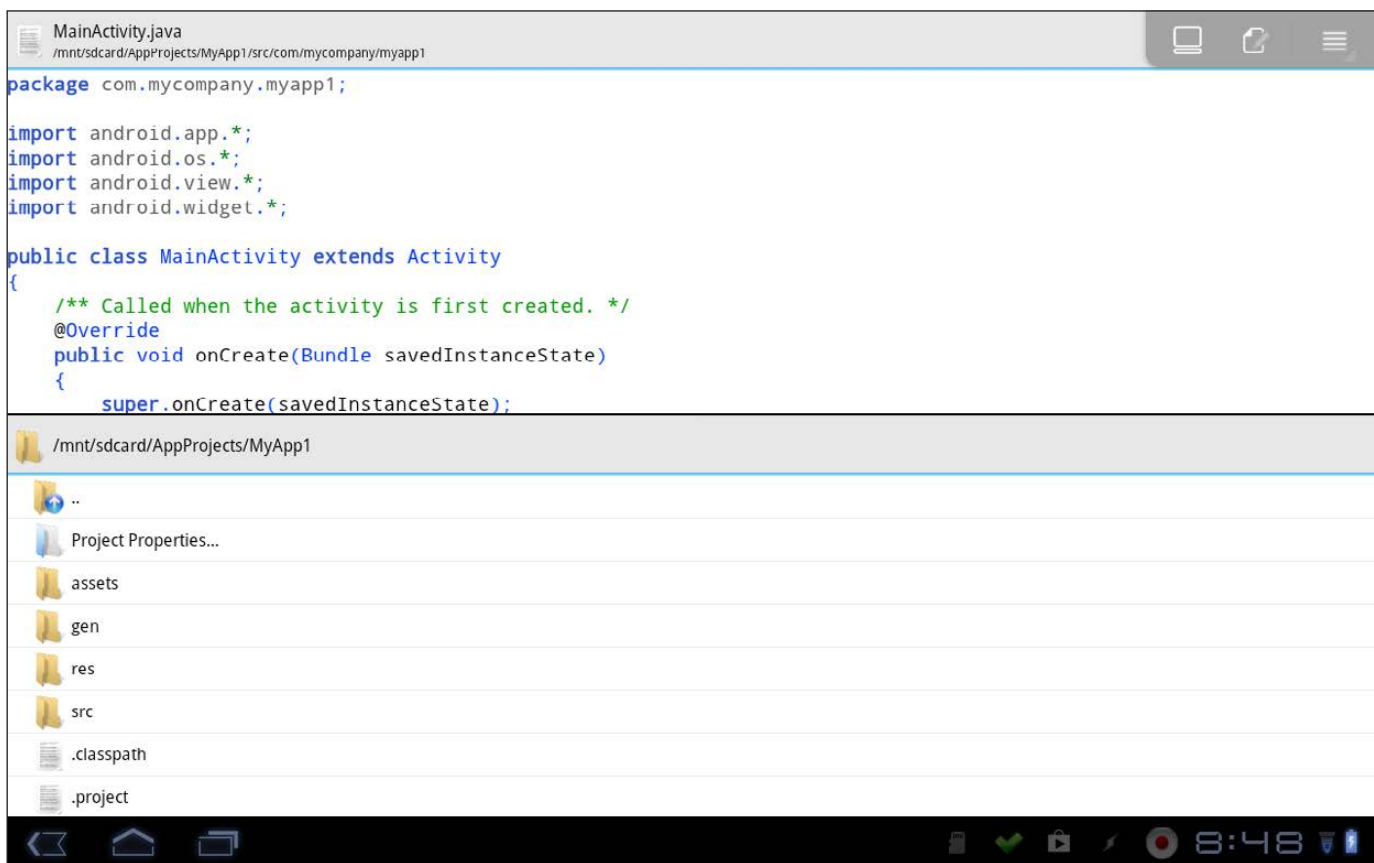


Figure 3. When you open a project, it gets pulled up into the IDE.

appname (where appname is the name you gave your project), and this new project will be opened up in the IDE. The main part of the IDE consists of two panes. Their alignment depends on the size of the device on which you're running it. On my phone, the panes are one above the other, and on my tablet, the panes are side by side. The first pane is a file listing for your project, containing all the properties, resources and source files needed for an Android project. The second pane is the main editor, where the central file (MainActivity.java) gets loaded on

project creation.

If you start with one of the templates, you can compile it and run it right away to see how the process works. Click on the menu button, and select the Run option. This will pop up a dialog, informing you of each step being done. It will compile your code, link it and create an APK file. In order to run it, this APK file needs to be installed. So an installation dialog will appear asking if you want to install it. Once installed, it then will start up, and you will have your very first Android application, developed

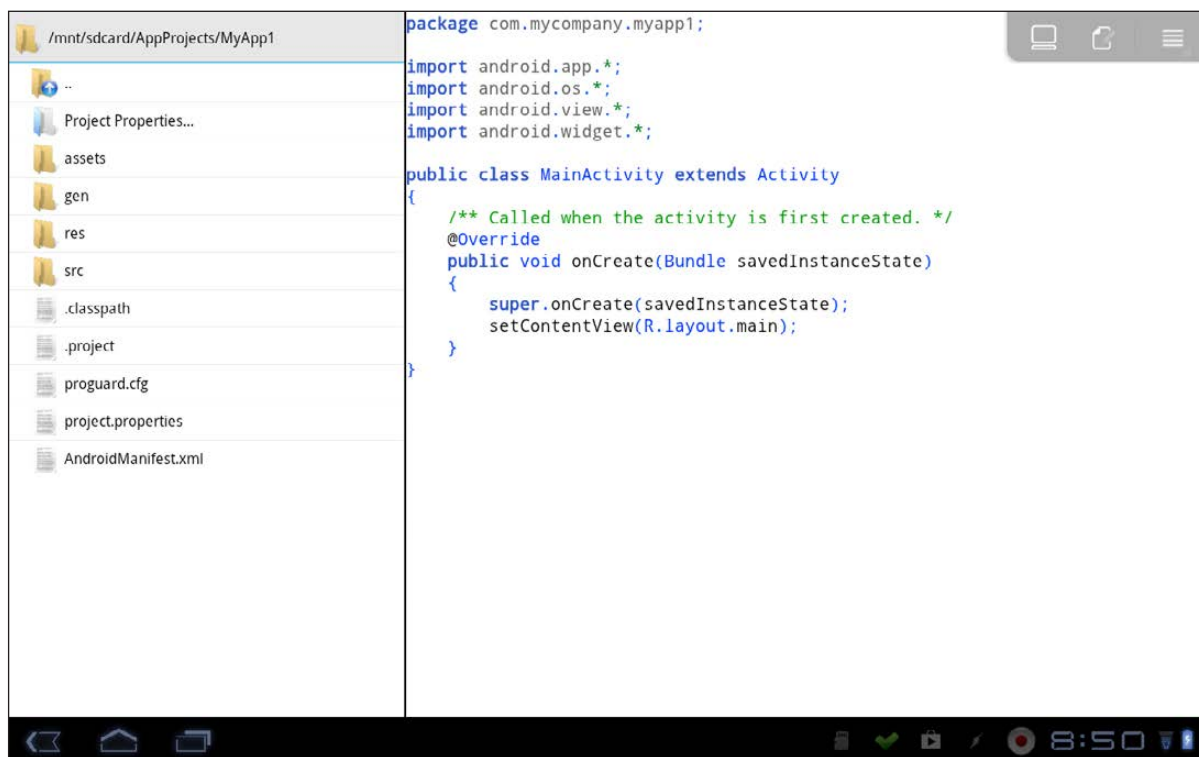


Figure 4. The tablet interface opens with panes side by side.

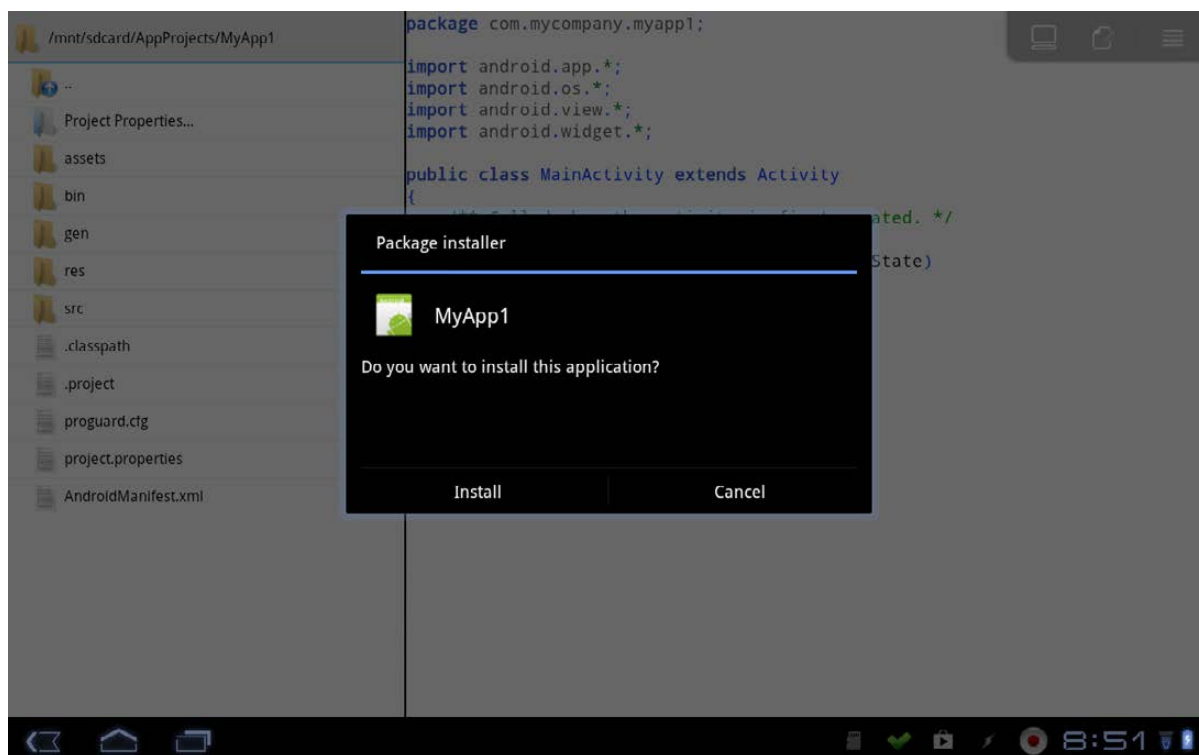


Figure 5. Security requires that you approve any app being installed.

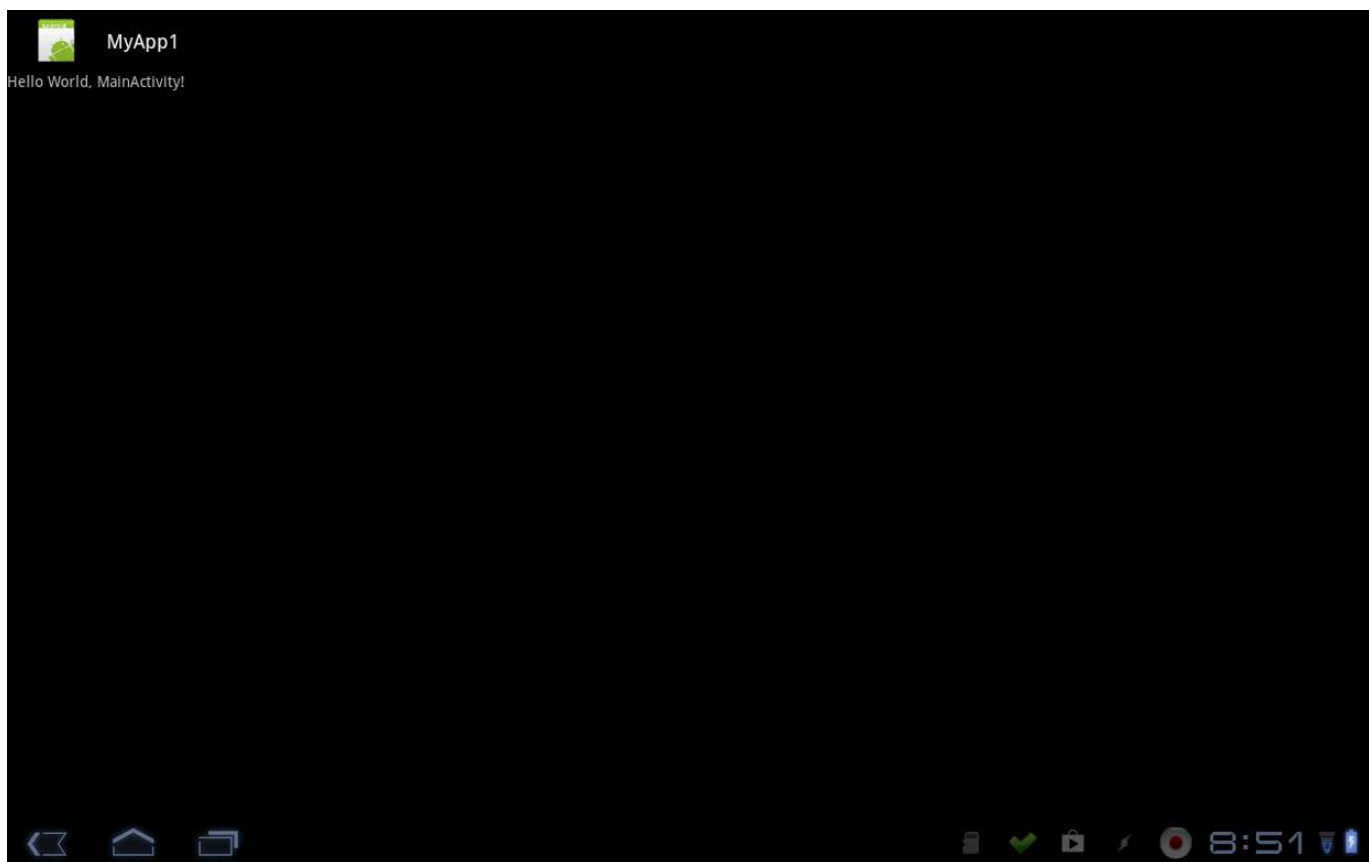


Figure 6. After compiling and installing, AIDE will run your new program.

completely on Android.

All of the apps that you develop in AIDE are structured as projects. Anyone used to using IDEs, like Eclipse, should be familiar with that. This means if you want to work on a different app, you need to close the current project and open a new one. To close a project, click the menu button, select More and then Close Project. This still will leave you in the main directory of your current project.

To see your other projects, click on the “..” entry in the file pane to move up one directory. Here, you should

see three standard entries, then a folder for each of your projects. The first entry is a tool to let you clone a Git repository to your Android device. The dialog that pops up allows you to enter a repository URL and a directory name. If you already have done some development work and have the code on Dropbox, you can download the relevant folder directly within AIDE.

The third option is to create a new project. Selecting this option brings up the dialog you saw when you started AIDE the very first time. To load a project into the IDE, it isn't

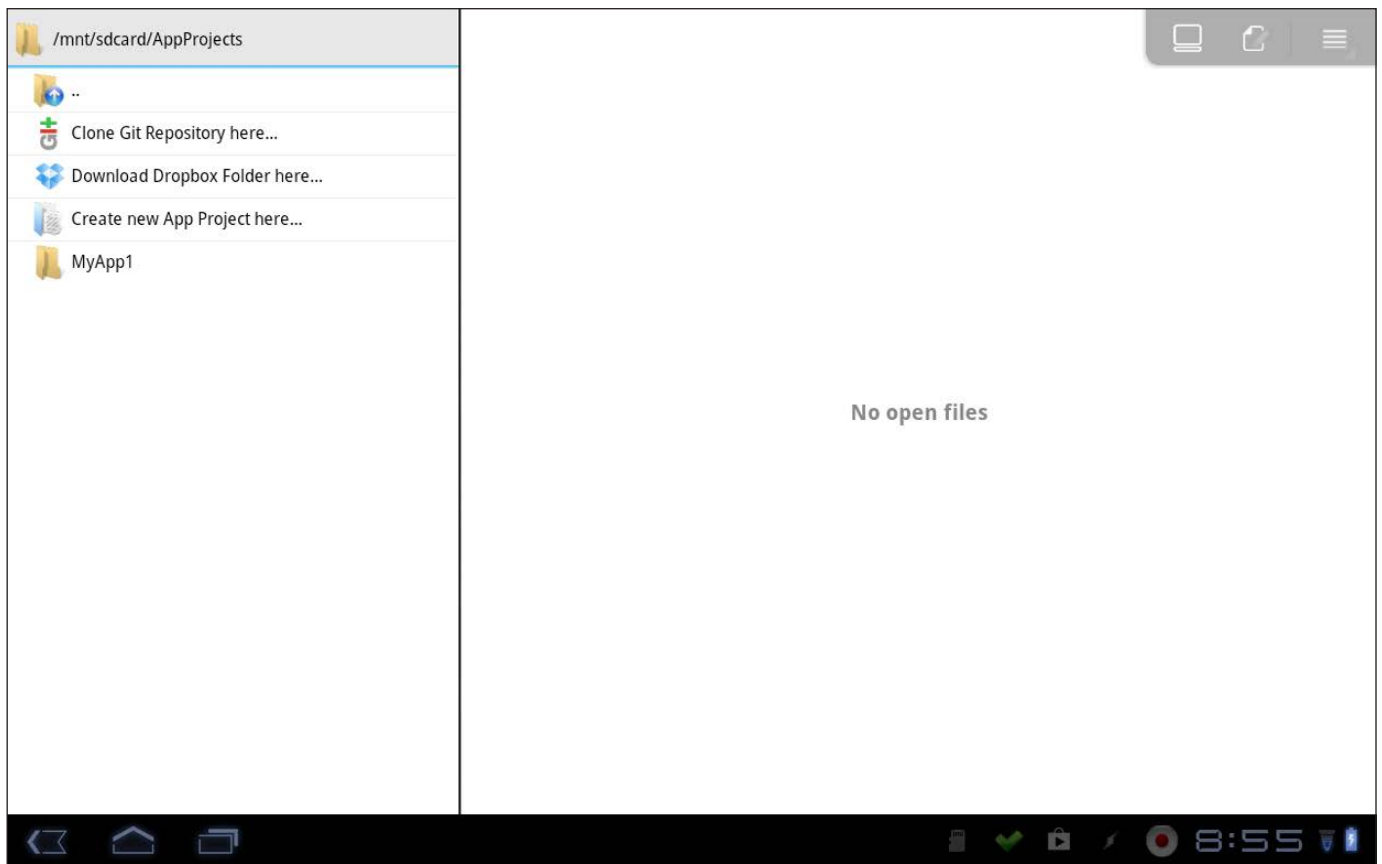


Figure 7. You can create new projects several different ways.

enough simply to select the project folder. Within the folder is an option to “Open this App Project”. This loads all the meta information about your project, like its properties and resource file locations.

Now that you know a little about how projects are handled, let’s start creating your very first app. If you have a project open right now, go ahead and close it. Change directories in the file pane until you are in the main directory for all of your projects. From here, you can select “Create new App

Project here”, and name your project “MyFirstApp”. You can set the package name to whatever you like, as long as it follows the usual format. Also, be sure that you have selected the “Hello World” app template.

Once everything is filled in, go ahead and click the create button. This will create all of the required files and open the new project in the IDE. The file MainActivity.java extends the class “Activity”. This is the main class you will be dealing with when developing apps. Your app builds on this class and overrides

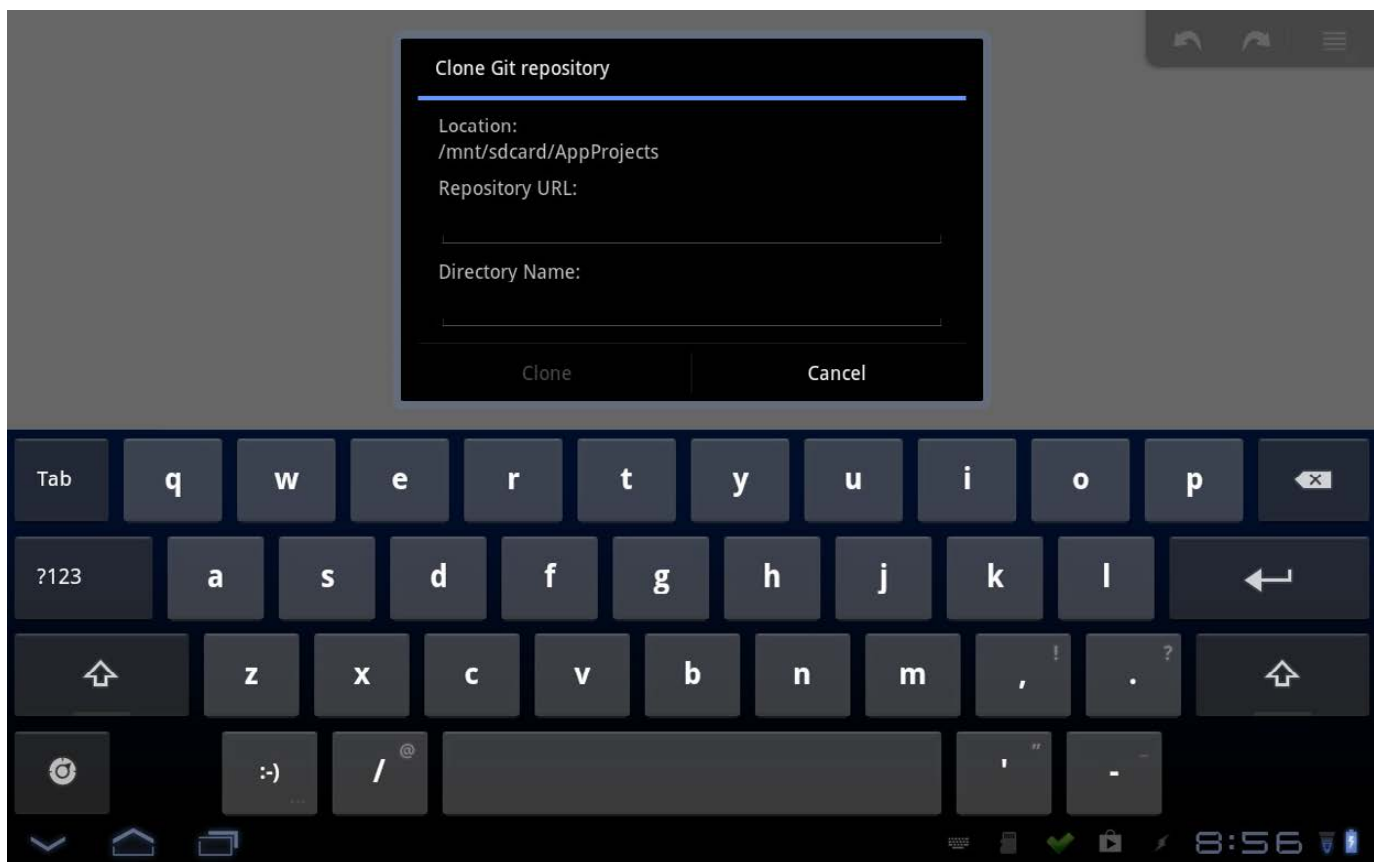


Figure 8. Cloning a Git repository is a fairly easy task.

the methods within it to make your app do its own thing.

The first method to override is “onCreate”. This method is called when your activity first is created. The usual tasks involve creating the graphical elements of your program and getting them initially rendered and displayed. The first step (super.onCreate) runs the code in the main class “Activity” to handle all of the OS-related tasks in creating a new activity. You then add your own code, usually GUI creation.

In the Hello World template,

it includes a call to the “setContentView” method, which assigns a particular layout as the content view. The beginning value is the layout “R.layout.main”. Layouts are XML files, containing all the elements that make up your graphical interface. To get to this file, you can change to the directory res, then layout. Within this subdirectory, you will find XML files for each layout that is defined.

In this simple program, you just have a single file called main.xml. Selecting it in the file pane loads it

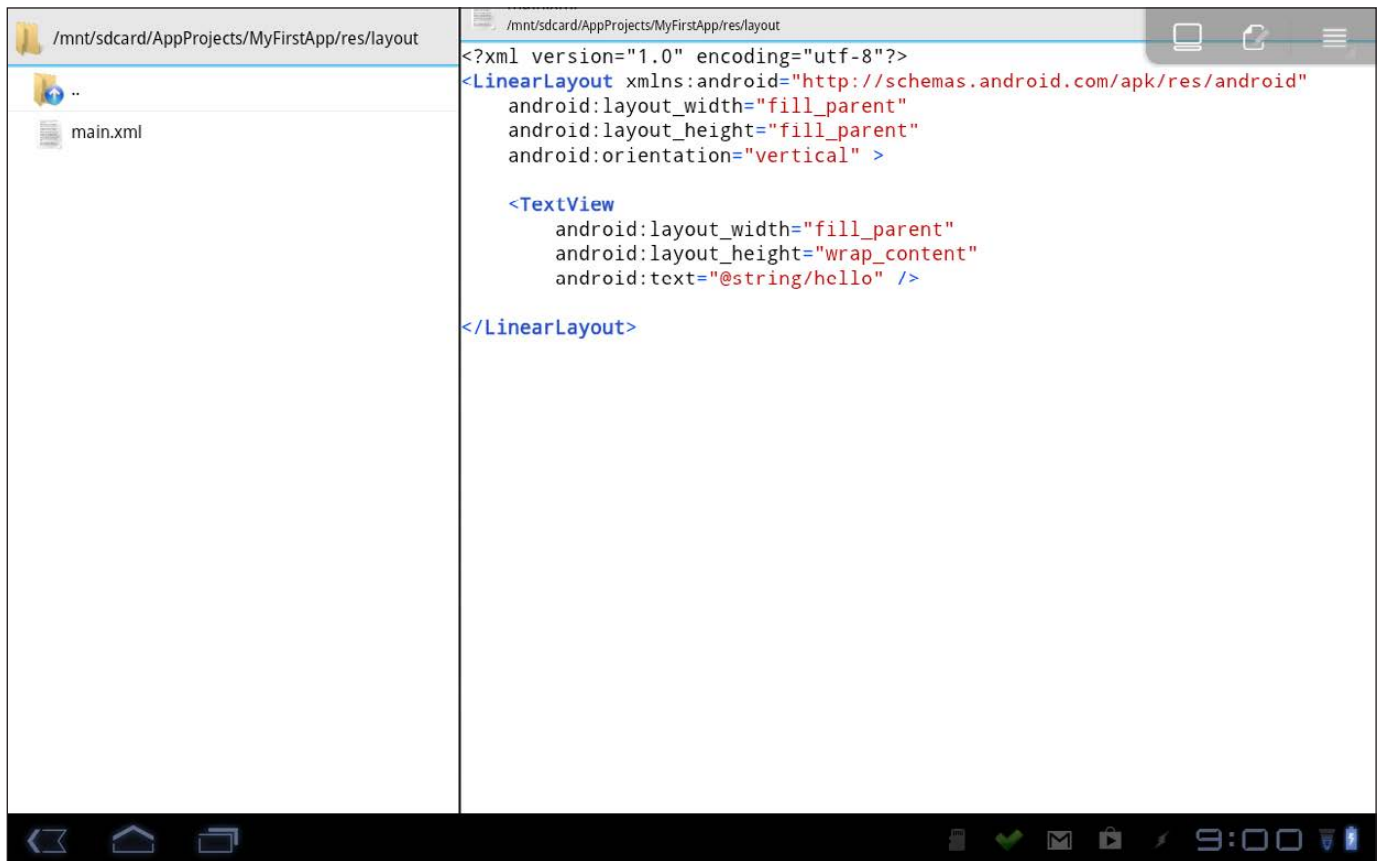


Figure 9. Layout files are stored in the subdirectory res/layout.

into the editor pane. The outermost container in the XML file is the type of layout being defined. A number of options are available. In the Hello World template, the layout type being used is a `LinearLayout`. You can set a width and height for the entire layout with the properties `android:layout_width` and `android:layout_height`. In this example, let's leave the defaults of "fill_parent" for both. You also can set the display's orientation; in this case, it is vertical.

The individual elements for the

graphical display are defined as internal XML containers within the outermost Layout container. In this example, there is a "TextView" object, where you can set various properties like the width, height or the text to be displayed. The proper way to include things like text strings, or anything else that may be reused, is to store them once as a resource and to refer to this string with a resource identifier. In this example, the identifier is "@string/hello". The actual text string is stored in the file `strings.xml`, which is located in the

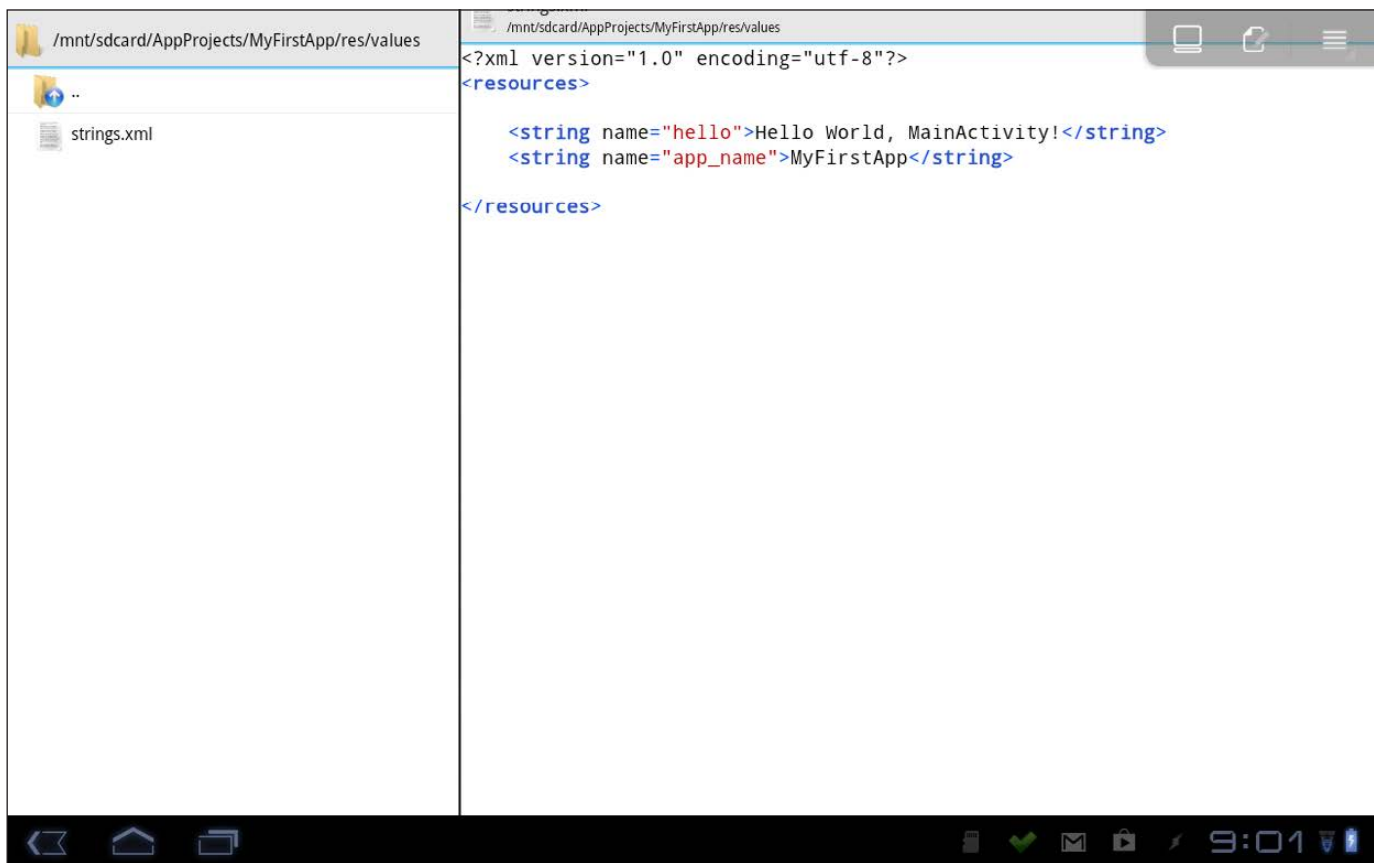


Figure 10. Resources for your program are stored in a series of XML files.

directory `res/values`. The outermost container of this XML file is the tag “resources”. Within this, is the list of all of the resources available to your program. In this case, there are only two strings: one for the app name and one for the text content of the main window.

To this example, let’s add a button to change the text being displayed. If you need to do any amount of typing, you will want to download and install a more complete soft keyboard or use an external Bluetooth keyboard. A good soft keyboard is the hacker’s

keyboard. It includes a lot of extra keys that are useful in code editing, including arrow keys to aid navigation and real control, Tab and Escape keys. Most soft keyboards included on Android devices lack these.

Open the main layout file, and below the `TextView`, add a new tag for a `Button` entry with the following code:

```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/button_title" />
```

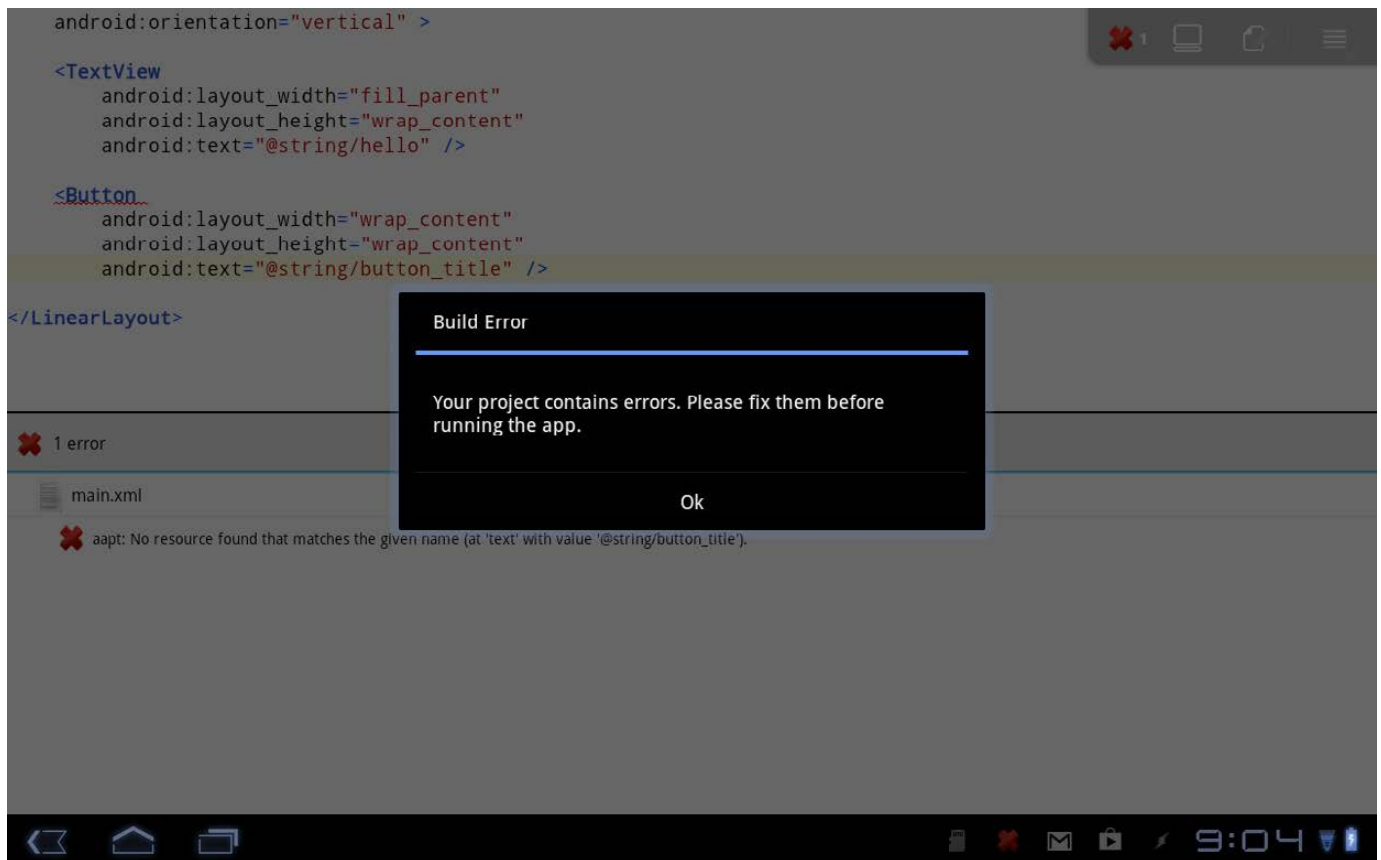


Figure 11. Any errors that crop up during a rebuild are listed here.

AIDE actually does a continuous code check to make sure there aren't any errors in the code. This means that while you are typing the above additions, you likely will see errors until you finally finish.

The editor includes tab completion, so you can start typing "<Butt" and then press Tab, and the editor will fill out the rest of the word for you. This is because "Button" actually is a special word in Android development.

Once you add the above, you can click the menu button, select More, and then select Refresh Build. This will

try to do a full rebuild of your app, and you will see an error about the missing string resource. If you have multiple errors, clicking on the error in the list will bring you to the location where the error appears. To fix this particular error, you need to add the following text to the file `res/values/strings.xml`:

```
<string name="button_title">My Button</string>
```

When you get to the end of this line and start typing "</", the editor automatically will fill in the rest of the line for you. When you refresh the build,

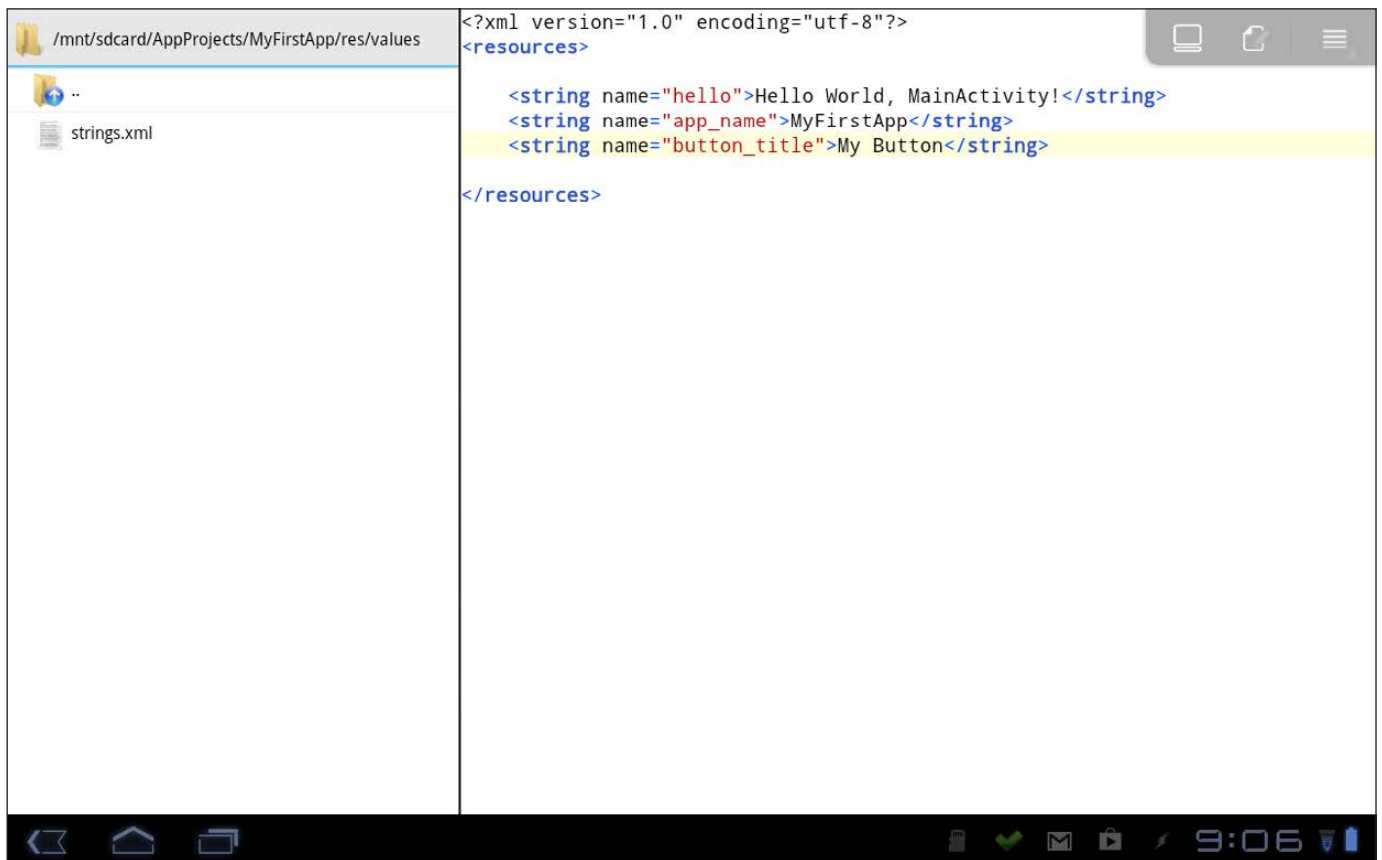


Figure 12. You can add new string resources to the file strings.xml.

the error should go away, assuming that you haven't introduced any typos.

Buttons are expected to trigger some reaction, however. This means you likely will want to add some kind of callback to a function in your button. This is handled within the layout file where the button is defined. You can add an extra property, "onClick", which gives a method name to be called when the button is clicked. For example, you might have the following in the button definition:

```
android:onClick="my_method"
```

You then can add the function "my_method" to the file MainActivity.java. This new method needs to be public and return void. Also, the only input parameter is a View object. Because you want to change the text in the TextView object, you'll need to add an ID so that you can refer to it. In the main.xml file, add the following property to the TextView entry:

```
android:id="@+id/view_text"
```

You then can use "view_text" to

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_title"
        android:onClick="my_method" />

</LinearLayout>

```

Figure 13. Add callbacks for buttons in the main.xml file.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

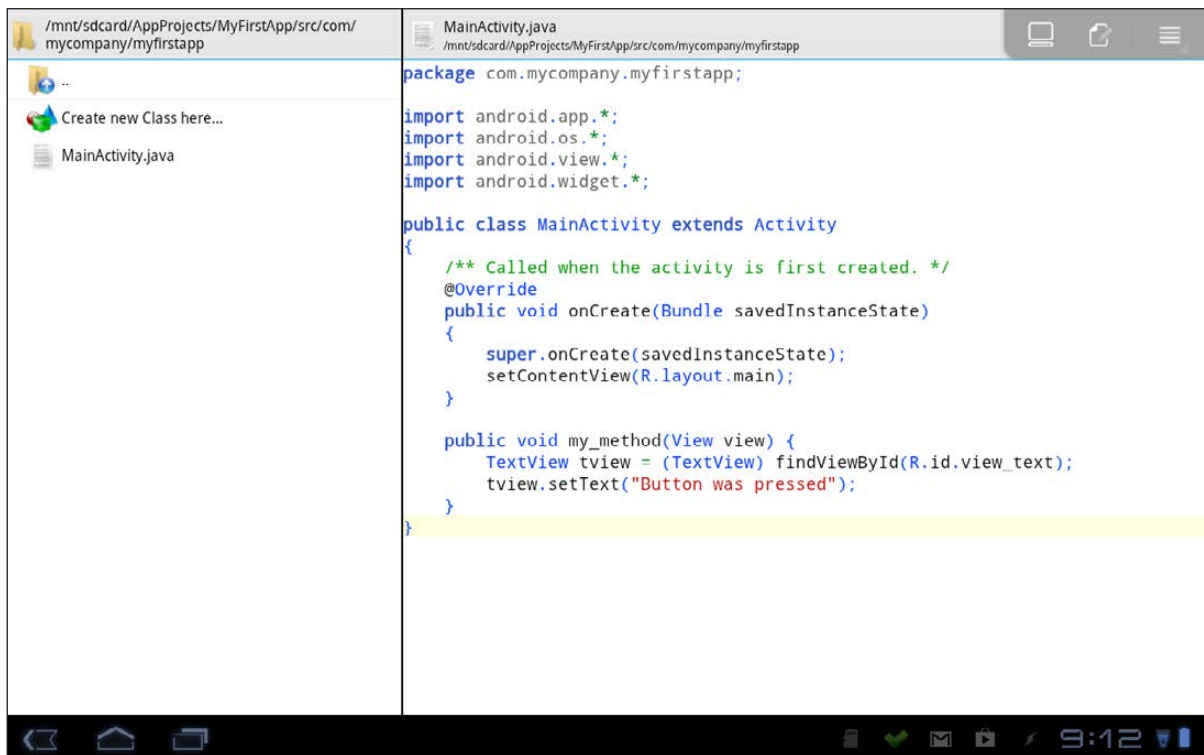
    <TextView
        android:id="@+id/view_text"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_title"
        android:onClick="my_method" />

</LinearLayout>

```

Figure 14. You need to add ID labels to interact with items in your program.



The screenshot shows an IDE window with a file explorer on the left and a code editor on the right. The file explorer shows the directory structure: /mnt/sdcard/AppProjects/MyFirstApp/src/com/mycompany/myfirstapp. The code editor displays the following Java code for MainActivity.java:

```
package com.mycompany.myfirstapp;

import android.app.*;
import android.os.*;
import android.view.*;
import android.widget.*;

public class MainActivity extends Activity
{
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }

    public void my_method(View view) {
        TextView tview = (TextView) findViewById(R.id.view_text);
        tview.setText("Button was pressed");
    }
}
```

Figure 15. The actual callback code goes into MainActivity.java.

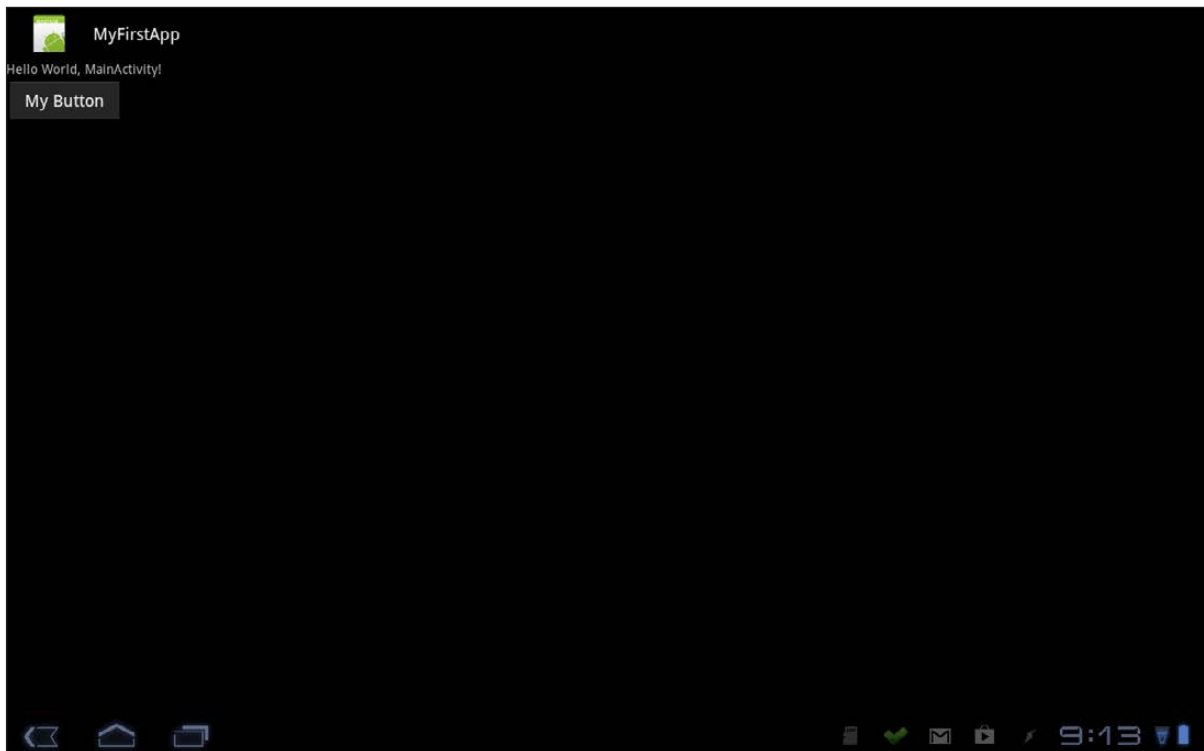


Figure 16. Before pressing the button.

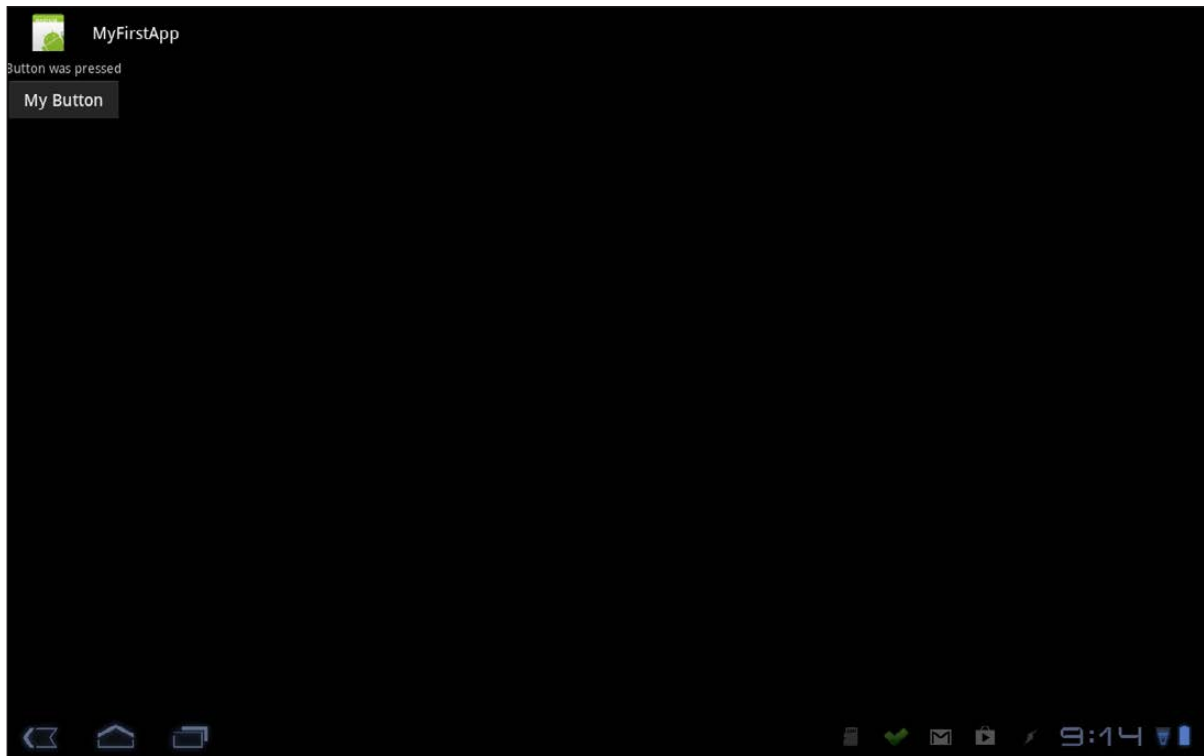


Figure 17. After pressing the button.

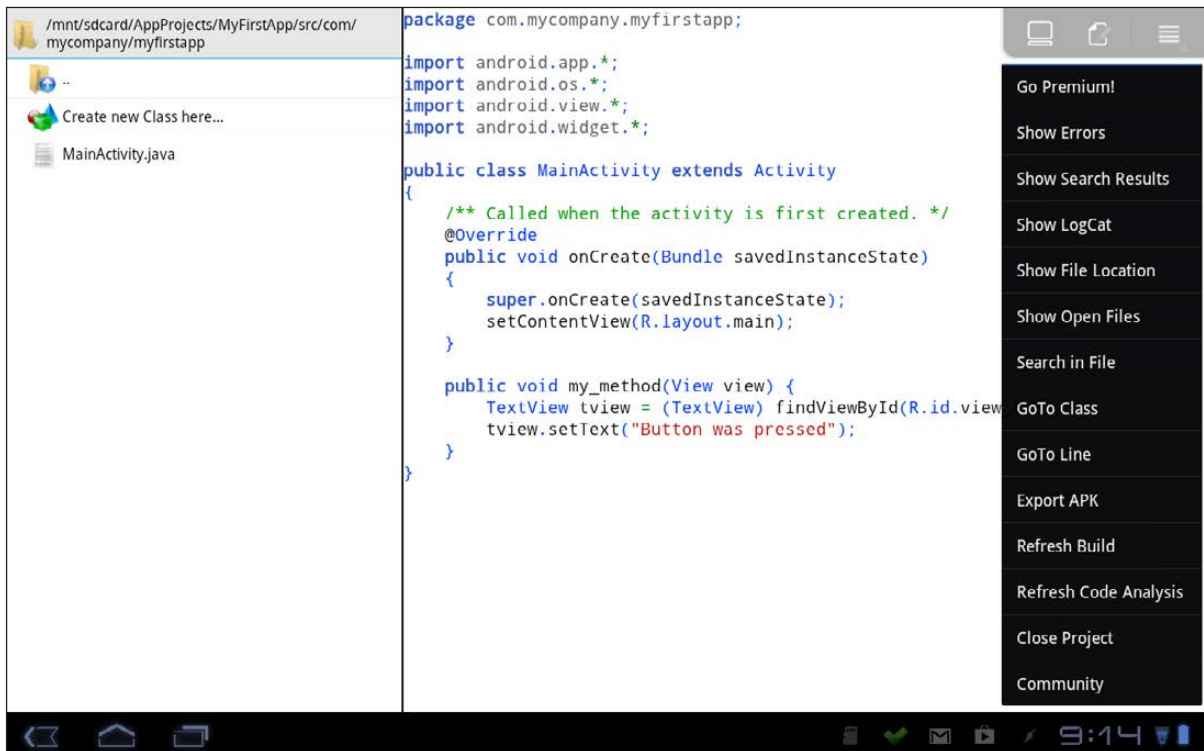


Figure 18. There are lots of functions in the menu not covered here.

COMPATIBILITY TEST SUITE

TEST YOUR ANDROID IMPLEMENTATION

**Porting Android to new hardware?
You *must* read this!**

NITISH TIWARI

Imagine having a refrigerator in your kitchen that could give you the nutritional contents and shelf life of the food you keep inside, or an automatic room cleaner that could take your commands via a chat app or SMS and clean your house in your absence. A few years ago, such devices were only dreamed of, but with rapid advancement in mobile operating systems and the ecosystem around them, such devices will be a reality soon.

Android specifically has an edge here. Because it is an open-source OS, developers all over the world are free to contribute, improve or even adapt it specifically to their needs (read: hardware requirements), and with the community growing larger by the day, much innovation is happening. But, this freedom to change the OS and adapt it to different hardware has to be managed somehow. There must be a mechanism to ensure that all the devices running Android provide the same experience to end users. Although developers get a chance to port Android to their hardware, users should not need to know about the underlying changes.

As per Google, Android compatibility

consists of three key components:

1. Android source code: the source code itself serves as a specification for the developers aspiring to port it to new hardware, and anyone who understands the code, will learn about the platform, APIs and their behavior.
2. CDD (compatibility definition document): CDD serves as the policy for Android compatibility. It is a document that aims mainly to clarify and remove ambiguity in the implementation. So if you want to port Android to new hardware, you should read the CDD and follow all the guidelines mentioned there.
3. CTS (compatibility test suite): this is the mechanism for Android portability. Because it is difficult—rather, impossible—to test hardware behavior via software, CTS helps with testing the APIs and the platform functionality. So, where CDD tells you what to do, CTS helps you check whether you have done it properly.

In this article, I discuss the CTS (Compatibility Test Suite), an application that allows you to validate your Android porting. CTS helps you

check that the app developers get the same set of APIs (and that those APIs will behave properly) and that users get the same overall Android experience in your new device. The testing is as easy as running test cases and generating a report and logs. Looking at the report will tell you whether everything is okay with your implementation. If that's not enough, you can send the report to Google to claim entry to Google play.

Overview

CTS, like Android, is an open-source tool, and the code for CTS is available with the Android source code. CTS also is available separately as an installable file on the Android developer Web site. So if you just want to run the tool without modifications, you can download it from the Web site. But, if you want to add or delete some test cases, or change a test case, you need to use the source code available with the Android source code, change it and then run the test cases. It is worth mentioning here that for every version of Android released, a corresponding CTS version is available. It's best to use the same versions of Android and CTS for testing. I use the latest CTS version 4.2_r1 for this article.

CTS is composed of two major

components: 1) the application that runs on your desktop and manages the test execution, and 2) the test cases that actually are executed on the mobile device or emulator. The test cases are written in Java as JUnit tests and packaged as .apk files, so that they can be run on the target Android device. Once the testing starts, the test application loads each of the .apk files (in the test scenario that is run) to the target, executes it, records the test result and, finally, removes it from the target device.

In this article, I provide a step-by-step guide to configure, run and generate the test reports. I also show how to add or remove test cases from the test harness. As an open-source enthusiast, I have an inclination toward the Linux OS, so I explain the CTS configuration on a Linux system first. But, Windows users need not worry, because I explain the configuration of Android on Windows also. Configuring it on a Mac machine shouldn't be much different, and once you configure it, running CTS is the same for all three platforms.

Prerequisites

To run CTS, you must have the CTS build downloaded from the Android developer Web site. If you are planning to add or remove test

cases, you need the CTS source code also, which is available with the Android source code. When you change the CTS source code, it has to be built again, and you will need a compiler for that, depending on your personal preference.

Along with the CTS, you also need to download the Android SDK available on the Android Web site. The SDK has the adb (android debug bridge) utility, which is required for CTS to run. Also, using the SDK lets you create virtual devices on which to run your CTS test cases if you don't want to use a real Android device.

Initial CTS Configuration

Here are the steps for configuring your Linux system to run CTS.

Ensure that adb is in the system path. This can be done with the command:

```
export PATH=$PATH:/path/to/android-sdk-linux_x86/platform-tools
```

Once the path is set, you can launch the CTS console by running the cts-tradefed script. To run the script, browse to the folder containing the android-cts folder and run the command:

```
./android-cts/tools/cts-tradefed
```

This takes you to the cts command prompt. Now you can run the test cases for your device using the cts commands. The commands are discussed below, but more details on configuring and running CTS are available in the official CTS documentation at <http://source.android.com/compatibility/downloads.html>.

If you use Windows, here are the steps to configure CTS on your computer.

Put the CTS prebuilt package downloaded from the Android developer site in a suitable location on your PC. Next, create a .bat file with the following contents and put it in the tools folder inside the CTS folder:

```
SET SDK_ROOT=<Path to the Android developer tools folder>
SET CTS_ROOT=<Path to the CTS folder>
if not exist %SDK_ROOT% GOTO showError
if not exist %CTS_ROOT% GOTO showError
SET PATH=%PATH%%SDK_ROOT%\platform-tools;
if not exist %SDK_ROOT% GOTO showError
java -cp
ddmlib-prebuilt.jar;tradefed-prebuilt.jar;
➔hoststestlib.jar;cts-tradefed.jar
-DCTS_ROOT=<Path to the CTS folder>
com.android.cts.tradefed.command.CtsConsole
:showError
Echo.
Echo "SDK_ROOT/CTS_ROOT not configured properly;
```



Figure 1. Android Virtual Device Window

Remember to change the values of `SDK_ROOT`, `CTS_ROOT` and the `-DCTS` root. This batch file sets the required environment variables and prepares your system to run the CTS.

Next, run the `.bat` file you created in the above step. This should create a `cts-tf` command prompt. Start your Android virtual device (not required for real devices), and run the command `list devices` at the `cts-tf` command prompt. This will list the Android devices that are available currently on your computer. If it doesn't, you may want check whether the previous steps were followed properly.

That's it for the CTS configuration, but if you are planning to use the virtual Android device, you need to configure it. To do that, just run the `SDK manager.exe` in the `adt` folder. This tool helps you manage the Android packages and the Android virtual devices. You can install your (modified) Android package and then create a virtual device with a target as your Android package. So, you can have a virtual device with your Android code loaded. (Skip this step if you plan to use an actual device, and just plug in your Android device to your computer via USB cable. CTS should be able to identify the device.)

Running CTS

The CTS testing mechanism is arranged as test cases, test packages and test plans. As the names suggest, test cases test a particular class, while the test packages are used to test functionality, which can be a combination of several APIs. Test plans are comprehensive testing schemes where features of the OS, such as signature and `appSecurity`, are tested. Now as you know, CTS is a combination of all these test cases and the application that manages these tests. So, to run a CTS test

```
cts-tf > list devices
Serial      State      Product   Variant   Build     Battery
emulator-5554 Available goldfish  generic   JB_MR1.1  0
```

Figure 2. Device Listing on the CTS Prompt

```
cts-tf > list plans
Android
AppSecurity
CTS-TF
CTS
Java
PDK
Signature
UM-TF
```

Figure 3. Plan Listing on the CTS Prompt

case, you need to communicate with the application, which internally will start running the tests.

The CTS application can be run via the `cts-fd` command prompt (which you get after running the `.bat` file mentioned previously). It has its own set of commands, and you need to use these commands to interact with it. The following are a few of the most important CTS commands:

- `help`: lists all the available commands and their options—type this command if you are stuck somewhere.
- `list`: the `list` command can be

used with different options to list the test packages, test plans, test results, devices and so forth on the command prompt. For example, if you want to list all the test plans, type `list plans`.

- `run cts`: this command runs the test cases in CTS. You can use it to run test packages, test plans and so on. For example, if you want to run the `android.view` package, type `run cts -p android.view`.

Now that you know the important CTS commands, you can start running the CTS test cases to verify your android implementation.

Report Generation

Once you run the CTS test case/package/plan, you need to wait for the test to finish running. Once the test is finished, CTS automatically generates the reports and logs. The reports generated by CTS are quite comprehensive with clear information about the passed/failed cases. If there are any failures in the test, the report shows it on the top along with the

Test Failures (3)		
Compatibility Test Package: android.view		
Test	Result	Details
android.view.cts.ChoreographerTest		
- testPostFrameCallbackWithDelayEventuallyRunsFrameCallbacksAfterDelay	fail	java.lang.AssertionError: Expected time delta less than 128000000 nanos, actually was 214605140 nanos. at android.view.cts.ChoreographerTest.assertTimeDeltaLessThan (ChoreographerTest.java:257)
- testPostFrameCallbackWithoutDelayEventuallyRunsFrameCallbacks	fail	java.lang.AssertionError: expected <2> but was <1> at android.view.cts.ChoreographerTest.testPostFrameCallbackWithoutDelayEventuallyRunsFrameCallbacks (ChoreographerTest.java:212)
android.view.cts.DisplayRefreshRateTest		
- testRefreshRate	fail	java.lang.AssertionError at android.view.cts.DisplayRefreshRateTest.testRefreshRate (DisplayRefreshRateTest.java:198)

Figure 4. Test Failure Summary in Report

Test Summary	
CTS version	4.2_r1
Test timeout	600000 ms
Host Info	[redacted] (Windows XP - 5.1)
Plan name	NA
Start time	Mon Mar 25 18:06:32 [redacted] 2013
End time	Mon Mar 25 18:26:57 [redacted] 2013
Tests Passed	627
Tests Failed	3
Tests Timed out	0
Tests Not Executed	0

Figure 5. Test Report Summary

failure details to help you analyze the failed test cases. You can access the report in the folder `android-cts\repository\results\<Time-Stamp>\testResult.xml`. CTS also generates detailed logs of the test case execution, which can help you if you need to troubleshoot issues. The log is available for both the device and

the host system. The logs are available in the folder `android-cts\repository\logs\<Time-Stamp>`.

Adding Tests to CTS

Thus far, I've been discussing how to run the default test cases available with CTS, but CTS is open source, so where is the fun if we don't play around with the code and add something extra? Let's look at how to write test cases for CTS and then integrate them with the framework.

Getting Started

Probably the best way to get started with adding a test case to the CTS is to look at the existing test cases present in the framework source code. (The CTS source code is available with the Android source code, so you need to download the Android source code.) As you go through the folder structure and the test cases, you will get an idea of

how the test cases are written and organized in the framework.

You can see that the default test cases included with CTS in the `cts/tests/` folder are segregated according to the different features they are meant to test. There are folders for `processtest`, `signaturetest`, `accessibility` and so on, which correspond to the various features. The folder `cts/tests/tests` has the general test cases, and most of the time, this is where you can add your own test case. If you go to a folder inside the `cts/tests/tests` folder, say the `bluetooth` folder, you can see it has an `AndroidManifest` file along with a `make` file. So, these tests essentially are Android apps written based on JUnit.

JUnit Introduction

JUnit is an open source, unit-testing framework for the Java language, which aims at testing the code and then its implementation. So, with JUnit, developers can add the test cases in their code and get feedback while developing the code. The Android CTS uses features like assertion and annotations in the test cases, so I primarily focus on these features here.

■ Assertion: using `assertion`,

you can test whether a given condition is true or false. JUnit provides an assertion method for all the primitive types and objects. So, you can use `assertion` in your test case to check whether things are as they should be. There also is an optional field for a string message that is output on failure. Assertion methods available in JUnit are `assertTrue`, `assertFalse`, `assertEquals`, `assertNull`, `assertNotNull`, `assertSame`, `assertNotSame` and so on.

■ Annotations: annotations are used before a method definition to identify whether it is a test method. They also are used to define the execution order of the test case or even ignore a test case. The annotations available in JUnit are `@Test`, `@Before`, `@After`, `@Ignore`, `@BeforeClass`, `@AfterClass` and so on.

Enough theoretical knowledge, let's look at a working sample! Let's take a working test case based on JUnit and see how it works. Below is a small code snippet from the CTS source to show how assertions are used in the test cases (you can see the full code here: `cts/tests/tests/`

bluetooth/src/BasicAdapterTest.java):

```
public class BasicAdapterTest extends AndroidTestCase {
    private static final int DISABLE_TIMEOUT = 8000; // ms timeout for BT
    disable private static final int ENABLE_TIMEOUT = 10000; // ms timeout
        ↳for BT enable

    private static final int POLL_TIME = 400; // ms
        ↳to poll BT state

    private static final int CHECK_WAIT_TIME = 1000; // ms
        ↳to wait before enable/disable

    private boolean mHasBluetooth;

    public void setUp() throws Exception {
        super.setUp();
        mHasBluetooth =
            ↳getContext().getPackageManager().hasSystemFeature(
                PackageManager.FEATURE_BLUETOOTH);
    }

    public void test_getDefaultAdapter() {
        /*
         * Note: if the target doesn't support Bluetooth at all,
         * then this method should return null.
         */
        if (mHasBluetooth) {
            assertNotNull(BluetoothAdapter.getDefaultAdapter());
        } else {
            assertNull(BluetoothAdapter.getDefaultAdapter());
        }
    }
}
```

As you can see, the `BasicAdapterTest` extends the `AndroidTestCase`. The first method in this class, the `setUp` method, sets a variable as true or false based on

whether the device has a Bluetooth feature available. The second method, `test_getDefaultAdapter`, tests whether an application can access the default Bluetooth adapter using the assert function available in JUnit. Similar methods to test different features are written. Additionally, you can use Eclipse to write new JUnit test cases—just select JUnit test case in the new Eclipse project window.

Integrate the New Test Case with CTS

Now that you know how to write a test case and where the files are stored, let's look at how to integrate these newly written tests to the CTS framework so they can be executed as the default tests. Following are the steps to add a test case to the CTS:

1. Develop the test case as a JUnit Android application. Add the folder to the `cts/tests/tests` folder.
2. Add the line `LOCAL_PACKAGE_NAME = <TestPackageName>` to the `Android.mk` file present in the application folder. Here, `TestPackageName` is the name of test package you just created.
3. Add the name of the test package to the `CTS_CASE_LIST` variable in the

MULTI-BOOTING THE NEXUS 7 TABLET

**Tired of Jelly Bean on your Nexus 7?
Try something else with MultiROM!**

BILL CHILDERS

Anyone who knows me well enough knows I love mobile devices. Phones, tablets and other shiny glowing gadgets are almost an addiction for me. I've talked about my addiction in other articles and columns, and Kyle Rankin even made fun of me once in a Point/Counterpoint column because my household has a bunch of iOS devices in it. Well, I was fortunate enough to add an Android device to the mix recently—a Nexus 7 tablet. I actually won this device at the Southern California Linux Expo as part of the Rackspace Break/Fix Contest, but that's a different story.

If you've not seen a Nexus 7, it's a nice little device. Like all "Nexus"-branded Android devices, it's a "reference" device for Google's base Android implementation, so it's got a well-supported set of hardware. I'm not trying to make this article sound like a full-fledged review of the device, but here's a few tech specs in case you're not familiar with it:

- 7" screen with 1280x800 resolution.
- 7.81" x 4.72" x 0.41" (198.5mm x 120mm x 10.45mm).
- 16 or 32GB of Flash storage (mine is the 16GB model).
- 1GB of RAM.
- NVIDIA Tegra 3 Quad-Core Processor.
- Wi-Fi, Bluetooth and optional 3G radios.
- Android 4.2 Jelly Bean.

The Nexus line of Android devices makes up the reference implementation for Android, so that tends to be the series of device that sees the fastest movement in terms of new builds of the OS, and in unique OS

derivatives like CyanogenMod. Right about the time I received the Nexus 7, Canonical released the developer beta of Ubuntu Touch, which targeted the Nexus 7 as its deployment platform.

Because I can't leave nice things well enough alone, I decided to start trying alternate OS ROMs on my shiny new Nexus 7. Ordinarily, each new OS would require you to reflash the device, losing all your configuration, apps and saved data. However, I found a neat hack called MultiROM that lets you sideload multiple ROMs on your device. How does it work? Well, let's walk through the installation.

Prep for MultiROM Installation

First, and I can't stress this enough, back up your device. I really, really mean it. Back up your device. You're messing around with lots of low-level stuff when you're installing MultiROM, so you'll want to have copies of your data. Also, one of the first steps is to wipe the device and return it to an "out-of-the-box" configuration, so you'll want your stuff safe.

Second, grab copies of the "stock" Nexus 7 ROMs as they shipped from the factory. You will want these in the event something goes wrong, or if you decide you don't like this MultiROM hackery and want to roll your device

back to a stock configuration.

Third, check the links in the Resources section of this article for up-to-date documentation on MultiROM. It's possible for things to change between this writing and press time, so follow any instructions you see there. Those instructions will supersede anything I type here, as this kind of hack can be a rapidly moving target. Also, do your own homework—lots of great YouTube videos describe this process, and a video sometimes can be worth several thousand words.

Notice: please make sure you follow these three steps, then follow the MultiROM documentation exactly. I'm not responsible if your tablet gets bricked or turns itself into SkyNet and goes on a rampage against humanity. Though I have to say, if that happened, it'd be kind of neat, in a geeky sort of way.

Unlocking Your Bootloader

Your device should be on the latest available factory ROM supported by MultiROM before you begin the installation. At the time of this writing, on my Nexus 7 (Wi-Fi-only) model, that was 4.2.2. The Nexus 7 comes from the factory with a "locked" bootloader. The first

thing you've got to do is unlock the bootloader before you can proceed.

To unlock the bootloader, you need the Android SDK tools installed on your computer (see the Resources section for a download link).

Specifically, you'll need the fastboot and adb tools for this, so make sure they're on your system and in your shell's PATH environment variable.

Next, hook up your tablet to your computer via the USB-to-MicroUSB cable, and then run:

```
adb reboot bootloader
```

Your tablet then will reboot, and you'll be in the Android bootloader. Once you're in the bootloader, run the following command:

```
sudo fastboot oem unlock
```

Next, you'll be prompted to confirm the command and accept that all data on your device will be erased. The tablet then will reboot, winding up in the setup wizard where you'll be prompted for all your setup information as if it were fresh out of the box once more.

Installing MultiROM

Now that your bootloader is unlocked,

the screen, then type the following on your computer:

```
sudo fastboot flash recovery recovery.img
```

That'll flash the modified recovery image that MultiROM requires to your tablet. Next, just tell the tablet to reboot by issuing the following command to it:

```
sudo fastboot reboot
```

Your Nexus 7 now is ready to install alternate ROMs.

Adding ROMs to MultiROM

Adding ROMs to MultiROM is fairly straightforward from here. Just hook your tablet up to your computer, drop the .zip file for the ROM you want to install onto the root of the filesystem, and then shut down the tablet. Restart your Nexus 7 in MultiROM by holding the "Volume Down" button while pushing the power switch. You'll see a screen with what appears to be the Android logo lying on its back (Figure 1). This is the bootloader. Push the "Volume Down" button until the red arrow at the top of the screen indicates "Recovery Mode", then push the Power button. This will boot the Nexus 7 into MultiROM.

Now that your Nexus 7 is actually in MultiROM, select the "Advanced" button in the lower-left corner, then select "MultiROM" in the lower-right corner. Now, to install a ROM, touch "Add ROM" in the upper-left corner (Figure 2).

Accept the defaults (unless you're trying the Ubuntu Touch developer release), and just press Next. The next screen will ask you to select a ROM source. Touch the Zip file button, then pick the .zip file of whatever ROM you want to install. The system will go ahead and install it, and it'll let you know when it's complete. Push the Reboot button when the install is complete, and your tablet will reboot into the MultiROM selection screen (Figure 3).

Looking at my boot menu, you'll see I've got cm-10.0.0-grouper installed, otherwise known as CyanogenMod. To boot that, I simply touch it, then press the large blue Boot button at the bottom of the screen. It's as simple as that—the Nexus 7 will just start booting CyanogenMod. At one point, I had the stock ROM, CyanogenMod, AKOP and Ubuntu Touch on my Nexus 7, all coexisting nicely (but they took too much of my limited 16GB storage space, so I pruned back some).



Figure 2. MultiROM “Add ROM” Screen

If you decide a particular ROM isn't for you, you can get rid of it quite easily. Just go back to the MultiROM install by booting with the Power and Volume Down buttons depressed, then select Recovery, and press the Power button again. Dive back into the MultiROM menus, just like you're installing a ROM, but instead of pressing Add ROM, press List ROMs.

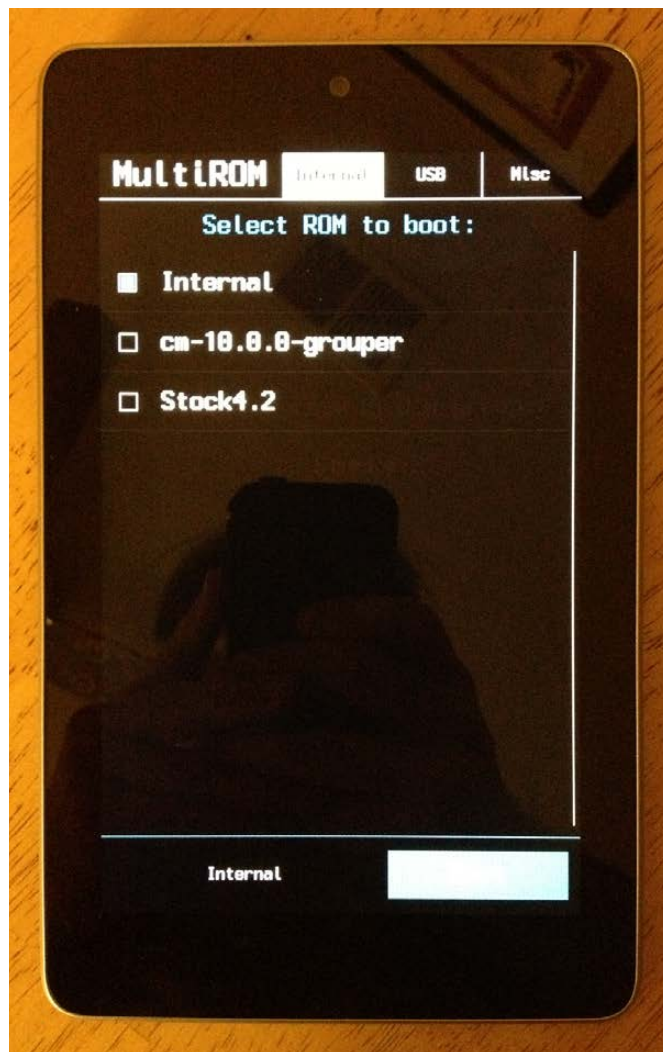


Figure 3. MultiROM Boot Menu

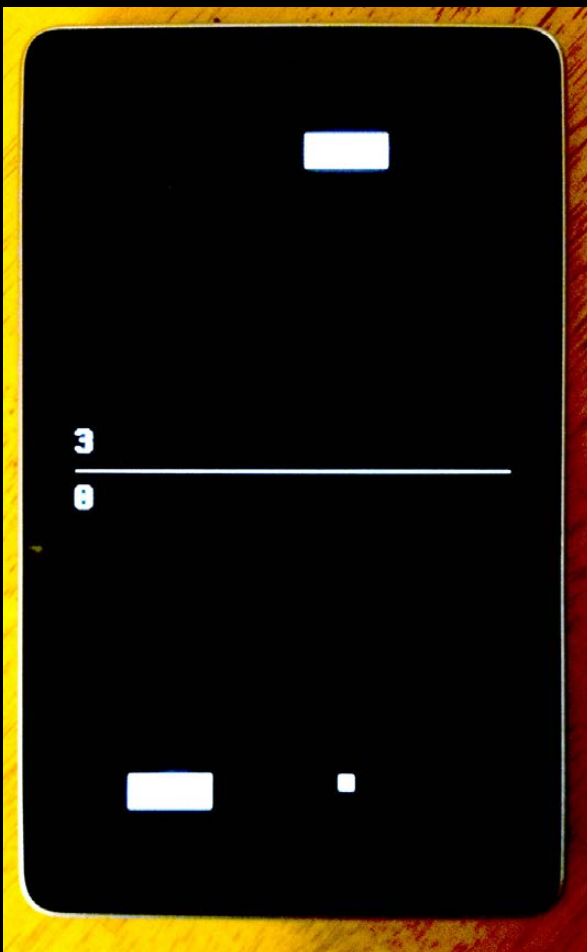
Touch the ROM you want to delete, and then just select Delete from the buttons that pop up. This will let you keep your MultiROM install clean, with only the ROMs you want to test active at any given time.

Getting Ubuntu Touch Running

Ubuntu Touch is something I've been watching closely, particularly

PONG

The programmer who wrote the MultiROM program has a great sense of humor, and he left a “Pong” easter egg in the software. From the main MultiROM boot screen, just touch the MultiROM logo, and you’ll get a proper portrait-orientation port of Pong (say that three times fast!)



Pong!

because I spent a little time with an Ubuntu Touch-equipped Nexus 7 at the Southern California Linux Expo. The Ubuntu Touch developer builds can be a little finicky, although they’ve stabilized in recent weeks. The key to getting them going in MultiROM is to select the “Don’t Share” radio button when adding the ROM (Figure 2). The Ubuntu Touch builds come in two parts. Add the smaller hardware-specific zip file first (on my Wi-Fi Nexus 7, it’s `quantal-preinstalled-armel+grouper.zip` on my Wi-Fi Nexus 7), but do *not* reboot—go back, list the ROM again, then push Flash Zip, and select the larger ROM file (`quantal-preinstalled-phablet-armhf.zip`). After that completes, you can reboot your tablet into Ubuntu Touch. Be advised, though, that Ubuntu Touch is under very heavy development, and sometimes the daily builds exhibit issues—and may not work at all. Your mileage may vary. If you do get Ubuntu Touch going, but it seems unresponsive to touch, try sliding from the left bezel toward the center. That’ll bring up a Unity-style launcher, and things should work from there. It took me a few tries to figure this out. I thought my Ubuntu Touch installation was broken or that I had a bad build.

Worms and Linux

Some people believe that Linux as an operating system inherently is not vulnerable to viruses and worms; others believe that Linux is not vulnerable simply because of its limited use as a desktop. The debate remains open.

HIMANSHU ARORA

Computer worms are considered to be one of the best weapons in a computer attacker's arsenal. Through these computer worms, evil-doers intrude into computer systems to destroy files, attack other computer systems, steal data and so on. The concept of a computer worm is not new; attackers have been using worms to attack computer systems for decades now. If you look back at the history of computer worms, you'll see that the computer worms that caused the most damage were directed toward the Microsoft Windows OS. Is this because of the number of Windows vulnerabilities, or is it merely due to the number of Windows users? The question remains unanswered. Meanwhile, apart from the Morris worm, very few worms have been directed toward Linux.

An Introduction to Computer Worms

A computer worm is a program that has malicious intent (that is, it can cause damage to a computer system). It can replicate itself and propagate over the network to infect other vulnerable computer systems. You also can think of a computer worm as a computer virus that requires no human action to start and that has the capabilities to self-replicate and traverse over the network.

A worm can arrive on a system through various means, such as an e-mail attachment, a downloaded file from the Internet, a file over a TCP/UDP connection from a trusted neighbor host or removable media like CDs, pendrives and so on. Once it has arrived on a system, it has three tasks to do:

- Hide from any system-monitoring software, like antivirus software.

The whole idea behind developing a worm is to cause maximum damage, and this can be possible only if the worm exploits one or more system vulnerabilities.

- Start the actual damage for which it is designed.
- Send a copy of itself to other hosts on the network to infect them too.

A worm that is able to do all these tasks effectively can be a real threat to a host or a network. Developing an effective worm requires a lot of system-/network-related knowledge, and this is the reason not every computer worm is successful.

Design a Computer Worm—Wearing an Attacker’s Hat

From an attacker’s prospective, three things need to be considered when designing a computer worm.

1) Exploiting system

vulnerabilities: The whole idea behind developing a worm is to cause maximum damage, and this can be possible only if the worm exploits one or more system vulnerabilities. Exploits are nothing but loopholes in the standard utilities or in the OS kernel that worms can use to get

the required power or privileges to compromise a computer system.

If a worm does not exploit a system vulnerability, the only way for a worm to compromise a system is to wait for the user to make some mistake. But, this requires writing a lot of code; thus, it is not considered a good design for developing a worm. Also, this design strategy does not ensure that a worm always will be successful in compromising and damaging the system.

On the other hand, if a worm exploits some vulnerability, such as a “zero-day” vulnerability (http://en.wikipedia.org/wiki/Zero-day_attack), it becomes very easy for a worm to compromise a system and cause maximum damage.

Note: some worms do not intend to cause any damage to the infected host (see my next point for details).

2) Stealth—the ability to remain undetected: This also is one of the major aspects of worm design. Even a worm with the deadliest intentions can’t do much damage if a system administrator or the monitoring

software that runs on a host can detect it easily. So, the ability to remain undetected on a system is yet another important factor that worm authors consider. Countless worms are developed each month, but only very few become potential threats because of most worms' inferior stealth capability.

The best way for a worm to stay undetected is to piggyback itself onto a trusted process by compromising it. A trusted process is a process running on a computer system that is deemed trustworthy by the system itself, and hence, it faces minimal interference from firewalls and antivirus software running on the system. A process can become trusted if it follows the relevant standards that are set by the US Department of Defense.

Worms have other ways of hiding on a system. For example, my article in the January 2012 issue of *LJ* discusses a small and not-so-destructive example of a piece of code that can exploit ELF executables on a Linux system: <http://www.linuxjournaldigital.com/linuxjournal/201201?pg=92#pg92>. The design of this malicious piece of code enables it to compromise the executables on a Linux system, and each time a compromised executable is run, many other executables are

compromised. Here, the malicious code remains on the system until all the affected executables are cleaned.

Some worms are designed to hide for a very long period of time on an infected host. Such worms do not intend to cause any damage to the host on which they reside. Rather, the sole purpose of these worms is to provide a back door (to each host on which they reside) to the worm author, who then controls the host remotely and activates the core of the worm at any given point of time in the future to launch DDoS attacks to a particular server. These types of attacks make it practically impossible to catch the actual attackers (worm authors) and generally are used to target government organizations and big corporations.

3) Speed and method of propagation: Every worm has a lifetime of its own before it is nailed down. That's the reason propagation speed is one of the top priorities for worm authors. Propagation speed directly depends on the method of propagation.

One propagation method worms use is via e-mail, where worms are transmitted in the form of e-mail attachments. If you look at the intricacies of this method, you'll find that propagation speed is not very high, because all the messages have to go through e-mail servers. This

adds an extra delay for a worm to reach its destination.

Another worm propagation method is to establish TCP connections directly. A TCP connection could be between an infected and non-infected host, or it could be between a non-infected host and a server where a fresh copy of a worm can be downloaded. Because TCP is a connection-oriented protocol, the three-way-handshake (http://en.wikipedia.org/wiki/Three_way_handshake#Connection_establishment), or the concept of acknowledgement, can cause unnecessary delays and hamper the worm's propagation speed. But, this propagation method is still faster than propagation via e-mail.

The fastest of all the propagation methods is over UDP connections. Because UDP is not connection-oriented and is not affected by network congestion, it offers speedier propagation for worms. The only drawback of using UDP is that delivery to the destination is not reliable, but that's not a real issue in the case of worms where more than one copy of a worm is sent to a host.

You may wonder why e-mail still is used as a propagation method if it has the slowest speed. E-mail provides a way for worms to reach a lot of

systems through contacts, while UDP makes sure that propagation is quick. So, there is no best way of impact using either e-mail or UDP. But, if you consider a method where the worm is sent through e-mail over UDP, the worm can propagate to a large audience in no time.

Detect and Counter a Computer Worm—Wearing an Administrator's Hat

Here are some methods for detecting a worm:

- Monitor trusted processes: you all know that non-trusted processes are monitored actively by programs like firewalls and antivirus software, and any deviation from the allowed behavior causes those processes to be quarantined and even interrupted. But, what about trusted processes? Although minimal, there are chances that a worm may attach itself to a trusted process. So, what happens in that case? Well, trusted processes also can be monitored to a certain extent. For example, each trusted process has specific work to do. If it can be monitored for any deviation from its work, worms attached to trusted processes also can be detected.

- Monitor IP address scanning: each process that tries to do some activity over the network tries to connect to an IP address. Although legitimate processes know the IP addresses (on which the connection needs to be established), worms try to scan all the IP addresses that fall in the host's IP address space. This means that worms would try to connect to all the IP addresses in the IP address space of the host, and any good monitoring software will detect these worms.
- Install honey pots on the network: a honey pot is a computer node on a network that is set up explicitly as a trap for intruders and malicious programs like worms. Honey pots let the intruder (or malicious program) do all the evil stuff while recording all its activities. This way, an alarm can be raised to all the nodes on a computer network, and the network administrator easily can block such unwanted activity on other nodes.

After detecting a worm, the first step should be to quarantine the process and strip off its permissions to execute code on areas like heap, stack and so on. If it continues to show the same suspicious behavior, affected

processes also can be killed. Other than this, by keeping your system updated (with all the latest patches to kernel and system utilities), you can minimize the threat.

Linux Worms—Why Are There Only a Few Successful Ones?

Worms on Linux have had a limited success rate. As I mentioned at the beginning of this article, some people believe that Linux as an operating system is inherently not vulnerable to worms, and others believe that Linux hasn't been very vulnerable to worms simply because of its limited use as a desktop. The following are some points I'd like to add to this debate:

1. Unlike Windows, most Linux users are not just end users; they know what they are doing on their Linux boxes. Linux users do not easily fall for traps like "Hey, your system has lots of viruses and trojans. Download AntiVirusFake.exe and get rid of this malware." Also, most Linux users keep their OS updated with the latest patches, which automatically keeps viruses, worms and so forth at bay.
2. The wall of permissions is the other factor that makes Linux a difficult place for viruses and worms. A

normal Linux user account does not have much privileges at its disposal, so a worm or virus cannot do much except affect the files and executables that are limited to a single account. Also, unlike Windows, you never are encouraged to log in as administrator or root and play around all day clicking on files and executables with root privileges. The Linux philosophy is that if you want to do something that requires root privileges, acquire root privileges for some amount of time, do the required work and then return to normal user mode.

3. Linux and even most of the software that is run on it is open source. A large number of people constantly are working to improve and polish Linux and other open-source software. This means that even if somebody is able to develop a virus or worm by exploiting a vulnerability in the Linux kernel or some popular piece of open-source software, it will take very little time for the community to fix and patch it. Bugs in the Linux kernel are fixed in no time.
4. Unlike Windows, Linux does not follow the “brain-dead” file

extension system. If a.out is an executable file on Linux, and if you try to run it, it will not run just because it has a .OUT extension. The executable bit has to be set on every executable in order to execute it. If you compare a scenario where a Windows user receives an e-mail message with an attachment named “HotBritneyWallpaper.exe”, just by clicking on the attachment, the virus/worm is installed in the system. On the other hand, in Linux, if an e-mail with a similar attachment is delivered, every user first has to save the attachment, then give execution permissions to the file in order for the file to execute. But, if the attachment boasts of just being wallpaper, why would even a normal Linux user provide the file with execution permissions? There you go—problem averted. Linux provides another layer of protection that prevents accidental execution of binary files.

5. Linux comes packaged in various distributions like Ubuntu, Red Hat, Cent OS and so on, and each distribution has its own set of software utilities. For example, several e-mail clients are used

across various Linux distributions, such as KMail, mutt, emacs, Mozilla Mail, Evolution, pine and so on. Now if an attacker writes an exploit for one of those, it's not going to impact all Linux users. So, unlike Microsoft, where an exploit for Outlook rings alarm bells all over the world, because Linux has more variety, this adds another layer of protection.

A Case Study of Some Linux Worms

In this section, I focus on two popular Linux-specific worms. Note: the Morris worm has been the most popular UNIX-based worm to date, but I don't discuss it here, because so much has been said and written about it already. Read http://en.wikipedia.org/wiki/Morris_worm for more information on the Morris worm.

The Linux Slapper Worm: The Linux Slapper is a worm that targeted Linux machines running an Apache server. This worm was detected in October 2002. The Slapper worm exploits the buffer overflow vulnerability of the OpenSSL software installed on the server.

The worm starts by scanning the Internet randomly for hosts, and when it gets one, it checks whether the host can be attacked. This is done by sending an incorrect HTTP GET request

to the target. If an Apache Web server is running on the target, the reply from the server would provide enough information for the worm to decide on the next step. After confirming that an Apache server is running on the host, the worm connects to port 443 (the SSL port) and tries to launch a shell (/bin/ssh) by exploiting the buffer overflow vulnerability of OpenSSL. This exploit specifically targeted Intel x86 platforms. Once the shell is launched, the worm sends a shell script containing an uuencoded copy of its own source code. Using the shell script, the source code is written to a file named .bugtraq.c in the /tmp/ folder.

Once the /tmp/.bugtraq.c file is saved, the worm depends on the availability of GCC on the host and Apache's permissions to execute the GCC compiler. If these requirements are in place, it compiles the .bugtraq.c file into an executable. This executable then is run with the IP address of the attacking computer as a command-line argument. This way, the worm propagates further by forming a peer-to-peer network. It also opens up a specific UDP port to listen and accept instructions to be carried out on/from the infected system.

The Linux Slapper worm had many variants. The A variant listens on UDP port 2002, while variants B and C

listen on 1978 and 4156, respectively. The B variant also acts like a virus and corrupts some other files by overwriting them with itself. It also sends data to a yahoo.com address. The C variant also listens to port 1052 periodically and provides a shell to the attacker who is connecting to this port. Besides these differences, all three variants used different names for the files saved in the /tmp/ directory.

Detecting and removing this worm is not very difficult. All you need to do is:

- Log in as root.
- Stop the Apache server.
- Go to the /tmp/ directory.
- Stop any process that is running with the same name as the worm executable file present in the /tmp/ directory.
- Remove the worm files (use `ls -la` to list the files).

Note: there is another Linux worm called DevNull that worked pretty much the same as the Slapper worm. DevNull also exploited the buffer overflow vulnerability in OpenSSL. It is sometimes referred to as Slapper variant D, but it's actually a different worm.

The Linux Lion (L10n) Worm:

The Linux L10n worm, also known as the Lion worm, was written by a Chinese hacker named Lion. He created this worm to warn the Japanese education department of controversial books being used in Japanese schools. According to the worm author, these books proclaimed the war crimes done by the Japanese against China and Korea as "legitimate".

The worm exploited a TSIG buffer overflow vulnerability in the BIND (Berkeley Internet Name Domain) server software. BIND server software is used to provide instructions to domain name servers (DNS) to convert Web addresses into IP addresses. The Lion worm scans port 53 of the class B network IP addresses, and if it finds a name server running on the target, it launches its BIND exploit on the target. If the exploit is successful, a copy of the worm is downloaded in form of a package. This package then is extracted and the startup scripts are executed.

The infected machines send the root passwords to a Web site hosted in China where the attackers tried to decrypt the password and gain administrative-level access to the infected systems. Reportedly, the

worm also created “back doors” for attackers to gain complete access to the infected system.

The Lion worm installed a rootkit in the infected systems in order to hide itself from being detected. This made system administrators’ jobs really difficult. The SANS (System Administration, Networking and Security) institute created a program known as “Lionfind” for sysadmins to detect whether the Lion worm is present on a suspected machine.

This worm installed the following files on a Linux system:

- /bin/in.telnetd
- /bin/mjy
- /bin/ps
- /bin/netstat
- /bin/lis
- /etc/inetd.conf
- /sbin/ifconfig
- /usr/bin/find
- /usr/sbin/nscd
- /usr/sbin/in.fingerd

- /usr/bin/top

- /usr/bin/du

And, to hide its footprints, the worm deleted these files:

- /.bash_history
- /etc/hosts.deny
- /root/.bash_history
- /var/log/messages
- /var/log/maillog

Experts believe that this worm cannot be removed, and infected systems have to be re-formatted in order to clean out the worm.

Antivirus, IDS and IPS Products for Linux

Linux is now the “most-used OS” on servers, and its popularity on desktops is growing day by day. With this increasing acceptance, Linux is gaining more attention from attackers and intruders. More and more viruses, worms, trojans and so on are being made for Linux these days. Although the success rate of Linux malware is low at the moment, this is no excuse for not

21st-Century DevOps—an End to the 20th-Century Practice of Writing Static Build and Deploy Scripts

Automating, standardizing and simplifying DevOps requires a model-driven process unchained from one-off back-end scripts.

TRACY RAGAN

Having served as a software developer since leg warmers and shoulder pads were in style, I've seen the distributed platform, from UNIX to Windows struggle, with the process of moving software changes from development to production over and over again. News from IBM and CA technologies about acquiring "Release Automation" solutions for the distributed platform has occurred in the recent past and has become in fashion once again. It seems our industry either enjoys the feeling of

déjà vu or simply chooses to forget what it already has done to address the problem of DevOps.

DevOps, at its core, is a process that simplifies the hand-off of source code between development and production, allowing test and production release teams to build and deploy binaries as required for the correct technology stack along the way. And, we have tools released in the not so distant past that have tried to solve this problem, but never quite met the challenge. Tivoli,

CA Unicenter and a host of other solutions from competing companies always have had the ability to perform software releases. But there is always something missing, and most of today's "in fashion" solutions do little different from their predecessors.

So what is missing? Is no one else noticing the 800-pound gorilla sitting in the room? The solutions that are paraded around as the latest-and-greatest method of solving the DevOps problem simply do not disrupt the

come for a 21st-century solution that is dynamic and model-driven for both build and deploy. Central to most DevOps tools is the claim of a "virtualized" process. However, these "virtualization" solutions choose to ignore the 20th-century static scripts that serve as the foundation of their solutions. A 21st-century "virtualized" solution cannot be built upon brittle, one-off scripts for either the build or the deploy. What is required is for developers to accept the paradigm

The solutions that are paraded around as the latest-and-greatest method of solving the DevOps problem simply do not disrupt the status quo enough to get to the core of the DevOps issue.

status quo enough to get to the core of the DevOps issue. They serve some ability to centralize logs and to manage server environments and configurations, which is certainly helpful. But they ignore the back-end one-off build and deploy scripts that contain the logic that actually does the work.

Yes, I understand. Ant and Maven are so cool that you may be compelled to tinker with them. But in the corporate enterprise, the time for tinkering is over, and the time has

shift that is needed for 21st-century DevOps, a model-driven, scriptless solution. And yes, change can be scary.

History can be our best teacher, and in the DevOps space, this is particularly true. Looking at how the UNIX administrators and mainframe administrators of the 1980s and 1990s addressed the problem can provide insight into what works and what does not work. On the mainframe, private compile and ship JCL was thrown out when "processors" were

introduced. Processors are the way that the mainframe creates dynamic build and deploy “scripts”, based on a model-driven framework. Everyone repeats and reuses the logic for compiling/binding/linking source code and shipping load objects. On the UNIX side, a central administrator, often using ClearCase, managed a central build and deploy script for the different levels and versions of the application moving across the life cycle. Even though they did not succeed in completely

releasing software. There is some hope, however. We may be seeing some movement away from scripting for releases, but few visionaries understand the build and tend to leave it out of the conversation.

So, what is in a build and why does it matter? Actually a substantial amount of information that is critical in the process of managing the DevOps effort is managed in the build. Get the build right, and a substantial amount of time and money is saved across the life cycle. Builds most commonly are

For reasons that are far beyond my understanding, it has been my perception that individuals who sell themselves as visionaries in this particular space avoid the discussion of scripts.

eliminating build and deploy scripts, they minimized the number of them down to a manageable level, often one build and deploy script for each environment—smart!

For reasons that are far beyond my understanding, it has been my perception that individuals who sell themselves as visionaries in this particular space avoid the discussion of scripts. Scripts are a huge bottleneck and a hidden cost in the activities of developing and

managed by static build scripts that are somewhat unintelligent. You pass it a set of commands, and it executes them starting at the top and ending at the bottom. What build scripts cannot do is everything that is needed for managing your DevOps process through release.

A software build needs the ability to be flexible and transparent in what and how it is building the software—for example, incremental builds, dependency management,

compile/link/archive options (debug vs. no debug), transitive dependency and the use of third-party libraries that make up the release target technology stack. Scripting languages lack the ability to manage these moving parts dynamically. As a result, you get redundancy—copied scripts for different needs and environments, and you often hear “it worked on my machine.” Scripts also cannot produce reports that provide the insight that allows validation of the binaries before

you spend on your release automation solution. The two are simply different sides of the same coin.

Deploy scripts are equally as critical. In a deploy script, the steps for performing the release should be super-standard, with no wobbles between one Java application using a Websphere server and another. Scripts are wobbly. It is their nature to be wobbly, as there are so many different ways to write a script. For any DevOps solution that claims

For any DevOps solution that claims automation and virtualization as the core features, one-off scripts delivered by the customer’s development teams should not be required to drive the automation tools.

a release. Scripts are black boxes that produce black-box binaries. At best, there is a guessing game as to what the script did, such as what libraries and options did it use to create the deployable objects. And, if you do not know what your build script did, you cannot guarantee consistent deployments, regardless of the level of virtualization you have achieved. A bad build absolutely will result in a bad deploy, no matter how much money

automation and virtualization as the core features, one-off scripts delivered by the customer’s development teams should not be required to drive the automation tools. When you are spending top dollars on a new DevOps solution, your developers should not need to deliver the foundation of that new solution. If you purchase a solution that requires build or deploy scripting, Ant, Maven, Make, Python or the next groovy-new-scripting



CLOUD WORLD FORUM

26th - 27th June 2013 • National Hall Olympia, London

www.cloudwf.com

Harness the Cloud opportunity in your business

FREE
EXHIBITION
Register Your
Place Now

Over 200 Speakers Including:



Dr Jeff Jaffe
CEO
World Wide Web Consortium



Francisco Garcia Moran
Director General
Informatics
European Commission



Dana Deasy
CIO
BP



Oskar Stål
CTO
Spotify



Eric van Miltenburg
Senior Vice President
YouSendIt



Jim Reavis
Co-Founder and
Executive Director
Cloud Security Alliance



Jeff Barr
Chief Evangelist
Amazon Web Services



Richard Harris
CIO
ARM



Sujay Jaswa
VP of Business
Development
Dropbox



Paul Coby
IT Director
John Lewis



Daniel Marion
Head of IT
UEFA



Tony McAlister
CTO
Betfair

2013 Sponsors Include:

Diamond Sponsor:



Platinum Sponsors:



Gold Sponsors:



Silver Sponsors:



Show Highlights:

1. Over 5,000 senior IT decision makers from around the globe
2. 200 visionary speakers – gain a unique insight from industry heavyweights and hear case study examples
3. 8 theatres with 150 seminars answering all of your cloud computing questions
4. 150 global exhibitors helping you discover the latest and most innovative IT products
5. Co-located with the Big Data World Congress – leading 2 day conference



DOC SEARLS

Android's Limits

Android is a lot more free than iOS, but there are limits. We need to break through those.

At its birth, Android was the horizontal and open solution to the problem of Apple's vertical and closed silo. On Android, hardware makers and software writers could build devices and apps, free to operate outside the walls of any vendor's closed garden.

This was fine, as long as we ignored the closed and vertical natures of three controlling forces in Android's market space: 1) mobile-phone companies; 2) Google's main business, which is advertising; and 3) every e-commerce vendor, each operating its own silo. So let's visit those, in order.

Mobile-Phone Companies

Before Android standardized a single popular platform for smart mobile devices, the phones we got were co-silo'd by partnerships between phone makers and phone companies. Back in the early 2000s, I sat in a

meeting where a parade of software developers presented groundbreaking ideas to Nokia, which had invested in a number of those same developers. At the end, one of the Nokia people explained that these were all interesting ideas, but that the company already had worked out plans for features to be rolled out during the next several years on the company's phones, in partnership with the carriers. This explained why, for example, my Nokia E62 lacked the Wi-Fi capability of the otherwise identical Nokia E61 (http://en.wikipedia.org/wiki/Nokia_E61). The E62 was built for carriers in the US, while the E61 was built for various European carriers. If AT&T didn't want Wi-Fi on the E62, it wouldn't be there.

We can thank Apple for driving a much harder and better bargain with AT&T than any other mobile-phone maker ever did. And we can



O'REILLY®

oscon

open source convention

JULY 22–26, 2013

PORTLAND, OREGON

oscon.com

Build your open source expertise

OSCON is the must-attend gathering of the best and brightest minds in technology, an opportunity to challenge your assumptions and spark your imagination. Join us for five immersive days of all things open source — new and innovative projects, major enterprise-wide deployments, and — from icons of the open source movement — deep perspective on where we've been and where we're headed.

OSCON 2013 Tracks

- Business
- Cloud
- Community
- Data
- Education
- Geek Lifestyle
- Java and JVM
- JavaScript and HTML5
- Mobile
- Open Hardware
- Operations
- Perl
- PHP
- Programming
- Python
- Software Architecture
- Tools and Techniques
- User Experience (UX)

**SAVE
15%
ON ANY PASS**

**USE CODE
LNXJRN**

O'REILLY®

thank Google for making sure that the smart mobile device market had a white-box operating system, so its hardware base could be as broad and generative as possible.

But those moves did not change the nature of phone companies, or our dependency on them. Back when JP Rangaswami (<http://confusedofcalcutta.com>) was Chief Scientist at BT (and I worked for him there as a consultant), he observed that the core competence of phone companies is billing, not telephony. This is the biggest reason (among many others) why no phone company ever would have welcomed, much less created, the Internet. After all, there is nothing in the Net's protocols that welcomes, much less creates, billing opportunities. But today most of the data paths between mobile devices and the rest of the Internet go through phone company connections, and those connections are built for billing. This is done through "plans" for data usage. In some cases (for example, Sprint in the US), there are plans that allow unlimited usage. But in most cases, customers take a plan that obligates them to pay for some number of MB or GB per month, with additional charges for going over "caps". While

phone companies need to make money by charging for services, it's an open question whether charging for data usage is the best, or the only, billing choice. It's just one that comes easily to a business that's already built to charge for minutes and text messages (the latter of which cost nothing on the open Internet).

Mobile phone systems are also walled in other ways the open Internet is not. For example, most data plans work only in the country where they are bought. Cross a national border and you risk "bill shock" for data use, which is charged at a much higher rate, even if you use the same company's wireless connections. While this looks like a bug to customers, it looks like a feature to the phone companies—and also to governments that enjoy getting a slice of the action and never liked the borderless Internet anyway.

Then there is the video issue. According to Lowell McAdam, CEO and Chairman of Verizon, video already makes up for half the data traffic on his company's wireless network, and will reach two-thirds by 2017 (<http://variety.com/2013/digital/news/verizon-topper-try-a-la-carte-to-protect-cable-1200338345>). Thus, the Internet is being turned into a distribution system for



Instant Access to Premium Online Drupal Training

- ✓ *Instant access to hundreds of hours of Drupal training with new videos added every week!*
- ✓ *Learn from industry experts with real world experience building high profile sites*
- ✓ *Learn on the go wherever you are with apps for iOS, Android & Roku*
- ✓ *We also offer group accounts. Give your whole team access at a discounted rate!*

Learn about our latest video releases and offers first by following us on Facebook and Twitter (@drupalizeme)!

Go to <http://drupalize.me> and get Drupalized today!



“content”. In terms of market power, content producers and distributors are sure to bias the growth paths of developers and their offerings, including Google and Android.

Google's Business

Back when Google began, in the late 1990s, advertising was barely imagined as a business model. In fact, Sergey Brin and Larry Page, Google's founders, initially were opposed to advertising. But since then, Google has become the largest company in the on-line advertising business, and advertising is the largest percentage of Google's revenue base.

Most of that advertising is not of the old Madison Avenue kind, which mostly sent brand messages out over print and broadcast media. Most of on-line advertising today, including Google's, is descended from direct marketing, which is descended from direct mail. It is meant to be personal, and it is guided by data provided through personal use, whether those persons like it or not. While Android does provide choices about privacy protection with each app one installs, users have little if any control over what happens to personal data that gets passed on to third parties, such as data brokers and analytics firms.

Lately the tide has been turning against privacy compromises and abuses by the on-line advertising business and the apps and services supported by it. Although Google takes a stronger moral stance on behalf of personal privacy than do many other on-line advertising companies, the fact remains that most users of Android phones and tablets are Google's consumers rather than its customers, which means those consumers are the product being sold to Google's actual customers, which are advertisers. From Steven Levy's book *In the Plex*: “We don't monetize the thing we create”, Andy Rubin says. “We monetize the people that use it. The more people that use our products, the more opportunity we have to advertise to them.”

Android phones also come defaulted with a pile of Google apps, placed in privileged positions in the phone's app directory. This does not sit well with Google's competitors. On April 9, 2013, Fairsearch, a coalition that includes Microsoft, Nokia and Oracle, filed a complaint with the European Commission against what it called “Google's anti-competitive strategy to dominate the mobile marketplace and cement its control over consumer Internet

data for on-line advertising as usage shifts to mobile”, adding “Google is using its Android mobile operating system as a ‘Trojan Horse’ to deceive partners, monopolize the mobile marketplace, and control consumer data” (<http://www.fairsearch.org/mobile/fairsearch-announces-complaint-in-eu-on-googles-anti-competitive-mobile-strategy>). That’s a bit hyperbolic, but I believe they have a case.

For our purposes in the Linux community, the main thing at issue is bias in the evolution of Android and the apps that run on it. As long as advertising remains Google’s main business, and Android remains a Google project, it will be hard to keep a bias toward advertising’s imperatives from having an influence.

E-commerce

In 1995, my wife asked a question so profound that it has haunted me ever since: “Why can’t I take my shopping cart from one site to another?” The answer is that every e-commerce site is a silo with its own shopping cart, its own cookies for visitors, its own pile of visitor data, and its own silo’d relationships with visitors and customers. Nearly two decades have passed since then, and the situation

Advertiser Index

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

ADVERTISER	URL	PAGE #
1&1	http://www.1and1.com	37
CLOUD WORLD FORUM	http://www.cloudwf.com/	121
DRUPALIZE.ME	http://drupalize.me	125
EMAC, INC.	http://www.emacinc.com	13
EMPEROR LINUX	http://www.emperorlinux.com	23
IXSYSTEMS, INC.	http://www.ixsystems.com	7
KINGSTAR USA	http://www.kingstarusa.com	42, 43
MANAGE ENGINE	http://www.manageengine.com	55
NEW RELIC	http://www.newrelic.com	2
O'REILLY OSCON	http://www.oscon.com/oscon2013	123
OVH	http://www.ovh.com	10, 11
SHAREPOINT	http://www.sptechcon.com	129
SILICON MECHANICS	http://www.siliconmechanics.com	3
USENIX	https://www.usenix.org/conference/fcw13	57

ATTENTION ADVERTISERS

The *Linux Journal* brand’s following has grown to a monthly readership nearly one million strong. Encompassing the magazine, Web site, newsletters and much more, *Linux Journal* offers the ideal content environment to help you reach your marketing objectives. For more information, please visit <http://www.linuxjournal.com/advertising>.



SPTechCon

The SharePoint
Technology Conference

August 11-14, 2013 • Sheraton Boston



The Best SharePoint Training in the World!

Choose from more than **90** classes
and tutorials, taught by the industry's
best experts!

Check out more than
55 exhibiting companies!



www.sptechcon.com

"SPTechCon covers just about every major SharePoint topic or issue out there. It's a great way to learn about the latest and greatest."

—David Sullivan, Knowledge & Information Manager, Altman Vilandrie & Co.

"If your goal is to learn something from a trusted leader in the industry and get an opportunity to network, you must go to SPTechCon."

—Brian Perryman, SharePoint Developer, Health South Corporation

"There were great tutorials. And every session I attended left me with concrete applications."

—Jeff Smothers, IT Specialist, AgChoice Farm Credit

"SPTechCon is a great event and well worth the admission fee."

—Chris Kauke, Systems Consultant, MassMutual

SPTechCon™ is a trademark of BZ Media LLC. SharePoint® is a registered trademark of Microsoft.

A BZ Media Event

