

FREE
DVD

GEOTAGGING
Add location data
to digital images

NOW FEATURING
LINUXVOICE

LINUX
MAGAZINE



LINUX

PRO

MAGAZINE

DECEMBER 2016

GEOTAGGING

Add GPS coordinates to your images,
and plot your images on maps

Gimp and Neural Networks

PowerShell in Linux

Could it really compete with Bash?

Cordova

Build mobile apps with
HTML5 and JavaScript

GNU Social

Social networking
meets Free Software



Greg Kroah-Hartman

"Total world
domination was
our goal."

Skype for Linux

Will the new version
fix old perceptions?

LINUXVOICE

FOSS Picks

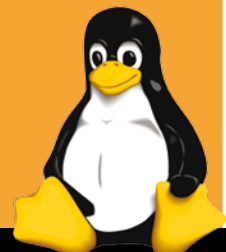
- Draw cartoons with OpenToonz
- VoxelShop retro 3D art

- Simon Phipps
- Vim 8
- NoSQL FAQ
- Gaming



Tutorials

- Ansible meets Docker
- Terminal multiplexing with Tmux



Issue 193
Dec 2016
US\$ 15.99
CAN\$ 17.99



HETZNER
ONLINE

More Power.

More Performance.

DELLEMC



Dedicated Root Server DX151

Dell PowerEdge™ R730 13th Generation
Single Intel® Xeon®
E5-2600 v3 (2.10 GHz) Octa-Core Haswell Processor
64 GB DDR4 ECC RAM, max. 192 GB at additional cost
Max. 8 hard drives at additional cost
50 TB traffic inclusive*
No minimum contract
Setup fee \$144.50

monthly from \$ **144**.50

Dedicated Root Server DX291

Dell PowerEdge™ R730 13th Generation
Dual Intel® Xeon®
E5-2600 v3 (2.10 GHz) Octa-Core Haswell Processor
128 GB DDR4 ECC RAM, max. 384 GB at additional cost
Max. 8 hard drives at additional cost
50 TB traffic inclusive*
No minimum contract
Setup fee \$211.50

monthly from \$ **211**.50

www.hetzner.de/us

* There are no charges for overage. We will permanently restrict the connection speed if more than 50 TB/month are used. Optionally, the limit can be permanently cancelled by committing to pay \$1.31 per additional TB used.

All prices include VAT and are subject to the terms and conditions of Hetzner Online GmbH. Prices are subject to change. All rights reserved by their respective manufacturers. Intel, Intel Logo, Intel Xeon and Xeon Inside are brands of the Intel Corporation in the USA or other countries.

SOMETHING NEW

Dear Reader,

If you took a glance at the cover before turning to this page, you probably already know our exciting news. But just in case you didn't notice, let me take this moment to welcome *Linux Voice* readers.

"Huh?" you're asking. Aren't you supposed to be competitors? Well yes, but the open source world has never fit neatly into anyone's definition of *doing business*.

The idea of merging our magazines came from a series of conversations, and the more we thought about it, the more we liked the idea. The *Linux Voice* team brings a passion, energy, and expertise that we truly admire, and, importantly, their approach is different enough from ours that we don't really step on each other's toes editorially. So why not give our readers both voices?

In the usual corporate publishing universe, if two mags in the same space merge, one of them takes over and the other one goes away. That happened to me earlier in my career when I worked for a small company that became part of a larger company. But we're not part of that usual corporate publishing universe, which gives us special powers, so we're using our special powers to find our own path. We like *Linux Voice*, they like us, and we think we can make enough room in our pages to let them keep doing what they do.

Is this an editorial choice? Are there other financial reasons for the change? Well that's not really my department, but I will say this: An editorial team is only part of a magazine.

Someone has to manage the circulation,

pay the bills, talk to the advertisers, and do all the business with printers and distributors. When you look at all the other factors that go into publishing, you start to see the benefits of small, like-minded publishers joining together to face the industry's dragons.

For me personally, the big excitement is welcoming such a distinguished and creative team of good writers. A warm welcome to

Graham, Ben, Mike, and Andrew, as well as the other *Linux Voice* contributors who are also part of this issue: Valentine Sinitsyn, Michel Loubet-Jambert, and Simon Phipps.

To our *Linux Pro* and *Linux Magazine* readers: I hope you're as excited about this change as I am. We're just getting started, but we're going to keep working to get it right, and I'm confident we can be the best of what we have always been and still offer you something new.

To our new *Linux Voice* readers: Come on in and get comfortable. We're excited to show you what we do, and rest assured, you'll find many familiar voices and faces.

A magazine is a living thing, with a personality, a history, and a future that are more than the contents of one issue. And as with other important things (humans, in particular) it is really good to mix things up once in a while – get out of the comfort zone and try something bold.

This is bold ..., and I hope you like it! We want to know what you think, so please send your comments and suggestions to: letters@linux-magazine.com.



Joe Casad,
Editor in Chief



LINUX MAGAZINE

WHAT'S INSIDE

This month's issue brims full with tools and insights on real-world Linux. A couple highlights:

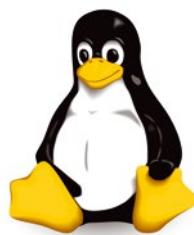
- **Deep Learning** – add color to your black and white photos using the power of a neural network (page 32).
- **Cordova** – this fertile framework from the Apache foundation lets you roll out apps for eight operating systems (page 50).

A special welcome to our friends in the Linux Voice community. Check out our new Linux Voice section, which starts on page 63.

SERVICE

- 3 Comment
- 6 DVD
- 96 Featured Events
- 97 Call for Papers
- 98 Preview

NEWS



08 News

- Linux 4.8
- Google Pixel phones
- Pi-powered Nextcloud Box
- Opera Sync Service password compromise

11 Kernel News

- Improving kernel locks
- `writetv()` error behavior
- Alerts from user space

14 Meet Greg Kroah-Hartman

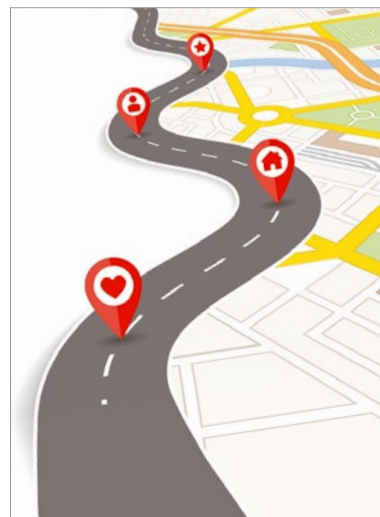
Get acquainted with the man behind the kernel stable branch.



COVER STORY

18 Geotagging

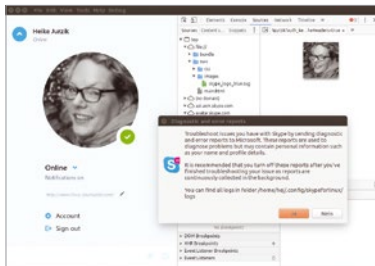
Add location data to your best digital images with digiKam and GPS Correlate.



REVIEWS

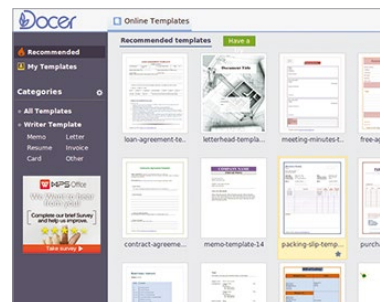
22 Skype for Linux

The old Microsoft gave the cold shoulder to Skype for Linux, neither updating it nor equipping it with new features. The new Microsoft promises to do better, with a brand new alpha prerelease version of a Linux client for the famous telephony tool.



24 Office Suites

In the office, the interoperability and cooperation of a few programs play an important role. We take the four big Linux office suites to task and see how well they cope with non-native formats.



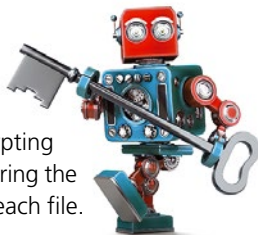
IN-DEPTH

32 Deep Learning

Deep learning isn't just for industrial automation tasks. With a little help from Gimp and some special neural network tools, you can add color to your old black and white images.

36 Cryptomator

Make files fit for the cloud by encrypting content and obscuring the name and size of each file.



40 PowerShell in Linux

Microsoft released its PowerShell under a free license in August 2016 and ported the tool to Linux and Mac OS. Is PowerShell for Linux a mere marketing ploy or a real contender that can compete with native Linux shells?

44 GNU Social

GNU social is a free and open source microblogging platform similar to

Twitter. You can join one of the public federated servers or set up your own in minutes.

48 Ask Klaus!

Klaus Knopper answers your Linux questions.

50 Cordova

Roll out an app elegantly and quickly for up to eight operating systems. According to the Apache Foundation, the only requirements are knowledge of HTML, CSS, and JavaScript.

54 cURL

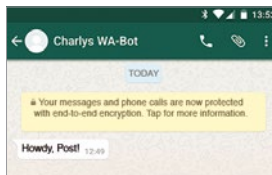
This powerful file transfer tool is simple to learn but can get you in trouble if you don't keep track of your options.

58 Termux

A terminal emulator that transforms an Android device into a lightweight and versatile Linux environment.

62 Charly – Outsmarting WhatsApp

Charly likes to keep up with the state of his domestic strawberry plants, so he has outsmarted WhatsApp to allow status messages to be sent through the service from a PC or Raspberry Pi.



LINUXVOICE

63 Welcome

Ben shares some thoughts on LV's new home.

64 Ownership and IoT

In the future your house will have lots of cool things, but don't think you'll own them.



65 Conduct Unbecoming

Why doesn't the Free Software Foundation have a code of conduct?

66 Editor Upgrade

Mike Saunders looks inside Vim 8 — the first major release of the text editor in 10 years.

72 NoSQL FAQ

Big Data is like the Matrix – Better without the sequel.

74 Core Tech: One ip to Rule Them All

Forget about arp and ifconfig; the ip command is a one-stop shop for network configuration and troubleshooting.

78 FOSSPicks

Draw cartoons with OpenToonz, build retro 3D images with Voxel-Shop, and explore the world with Marble 2.0.



84 Gaming on Linux

Try your luck with Rocket League, Fear Equation, and Master of Orion.



86 Tutorial: Autodeploy Ansible Containers

Streamline software deployment with Ansible and Docker containers.

92 Tutorial: Multitasking with Tmux

Operate multiple terminal sessions in a single window.

On the DVD

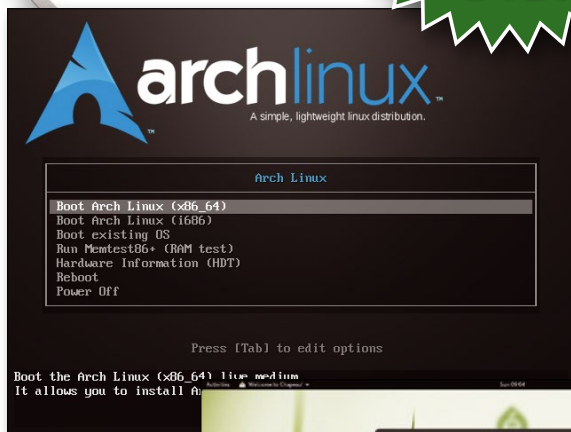


Arch Linux

The Arch community bills their distro as “a lightweight and flexible Linux distribution that tries to Keep It Simple” [1]. Although generally considered a distribution for Linux mavens [2], community members maintain a Newbie Corner for the curious [3]. The rolling release on this DVD is dated 2016.09. Pacman manages binary packages from official and community repositories, and the unofficial Arch User Repository (AUR) offers user-made PKGBUILD scripts for packages not included in the repositories.

Chapeau 24

Chapeau starts with Fedora Workstation and adds drivers and codecs missing from the all-free Fedora. The goal of the Chapeau developers is to create a friendly, out-of-the-box Linux that is for Fedora what Ubuntu and Mint are for Debian. This Live boot disc [4] derives from Fedora 24, with Linux kernel 4.7, Gnome 3.20 desktop, and the Yumex-DNF package management system. Chapeau is packed with audio, video, and productivity offerings [5]; version 24 also includes Nmap, Wayland, and five lightweight games.

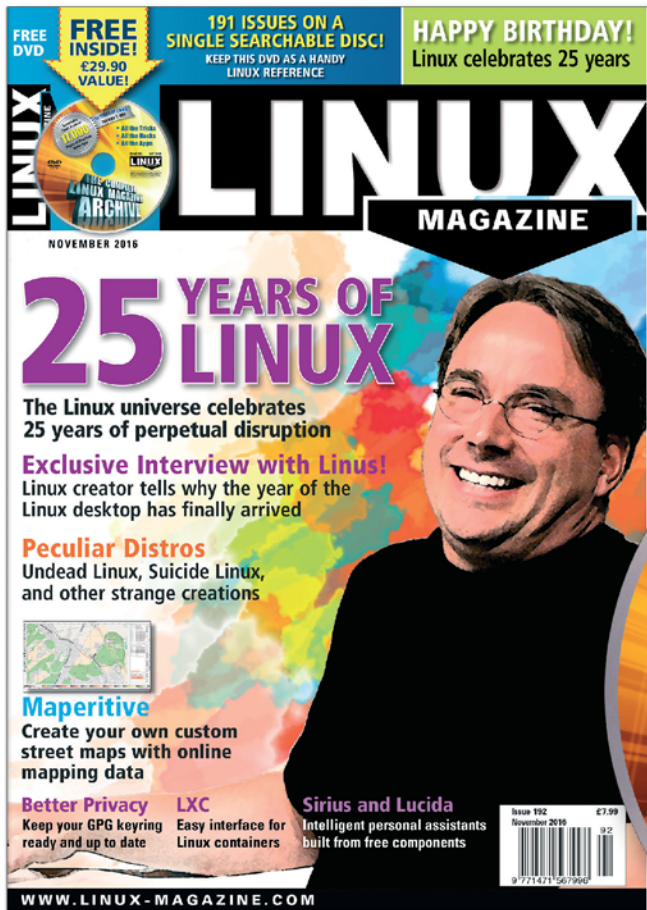


ADDITIONAL RESOURCES

- [1] Arch Linux: <https://www.archlinux.org>
- [2] ArchWiki: <https://wiki.archlinux.org>
- [3] Arch Linux Newbie Corner: <https://bbs.archlinux.org/viewforum.php?id=23>
- [4] Chapeau 24: <http://chapeaulinux.org>
- [5] Chapeau 24 features: <http://chapeaulinux.org/chapeau-24-features/>

Defective discs will be replaced. Please send email to subs@linux-magazine.com.

Happy 25th Anniversary to Linux!



Help us celebrate 25 years of Linux with the All-Time Archive DVD of Linux Magazine!



Order today and get all 191 issues of Linux Magazine on one handy DVD – FREE with Issue #192!

BEST VALUE: Subscribe to the print edition and get the archive DVD FREE with November 2016 issue (a \$39.90 / €29.90 value). You'll get Linux Magazine every month and 15+ years of content.

Order Now! Shop.linuxnewmedia.com

NEWS

Updates on technologies, trends, and tools

THIS MONTH'S NEWS

08 Torvalds Announces Linux 4.8

- Linus Torvalds Announces Linux 4.8
- Google Announces Its Own Pixel Phones

09 Nextcloud Announces Rasp Pi Cloud Box

- Nextcloud Announces Raspberry Pi-Powered Home Cloud Box
- Linus Torvalds Confirms Date of the First Linux Release
- More Online

10 No Headphone Jacks in New iPhones

- It's Official: No Headphone Jacks in New iPhones
- Opera's Password Sync Service Compromised

Linus Torvalds Announces Linux 4.8

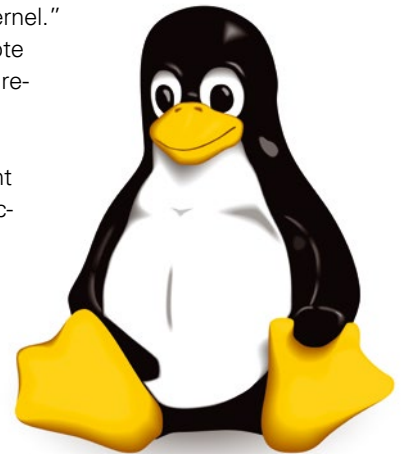
Linus Torvalds announced a new version of the Linux kernel with many notable new features. The two most interesting features of the Linux 4.8 release are support for Microsoft's Surface 3 touchscreen and the Raspberry Pi 3's SoC (system on chip).

Ubuntu 16.10, which is slated to be released in October, will be the first major desktop Linux distribution to use the 4.8 kernel.

However, not everything went as smoothly as expected. Torvalds noticed a nasty bug that slipped into the final release. He wrote on the mailing list: "I'm really sorry I applied that last series from Andrew [Morton] just before doing the 4.8 release, because they cause problems, and now it is in 4.8 (and that buggy crap is marked for stable too). In particular, I just got this "Kernel BUG at ./include/linux/swap.h:276" and the end result was a dead kernel."

Torvalds is not particularly happy about it and wrote that "Yes, I'm grumpy. This went in very late in the release candidates, and I had higher expectations of things coming in through Andrew. Adding random BUG_ON(s) to code that clearly hasn't had sufficient testing is *not* acceptable, and it's definitely not acceptable to send that to me after rc8 unless it has gotten a *lot* of testing, which it clearly must not have had. Adding stable to the cc too to warn about this."

Torvalds' advice to seasoned kernel developer Andrew Morton is to stop taking those kinds of patches!



Google Announces Its Own Pixel Phones

Google has announced the new Pixel family of smartphones powered by the Linux-based Android operating system. Previously Google partnered with hardware vendors like HTC, LG, and Samsung to sell the Nexus range of devices. With Pixel, Google will have total control over the design and hardware components of their smartphones. Google already sells Pixel Chromebooks and the Pixel Android tablet.

There are two models of Pixel phones: The Pixel showcases a 5-inch display and the Pixel XL features a 5.5-inch display.

Both Pixel phones have identical hardware, except for display size and battery capacity. The Pixel XL showcases a 5.5-inch AMOLED QHD (2560x1440) 534ppi display, whereas the Pixel features a 5.0-inch AMOLED Full HD (1920x1080) 441ppi

display. The Pixel XL comes with a 3,450mAh battery, and the Pixel comes with a 2,770mAh battery.

Both Pixel phones are powered by the Qualcomm Snapdragon 821, 2.15GHz + 1.6GHz, 64-bit quad-core



processor. The Pixel XL and Pixel come with 4GB of LPDDR4 RAM. Both models are available in 32GB and 128GB capacities. The Pixels will be running Android 7.1 Nougat.

Google claims that the Pixel phones have the best smartphone camera. In a blog post, Google said, "Pixel has a 12.3MP camera, featuring an f/2.0 aperture and big 1.55 micron pixels to capture lots and lots of light."

One of the most interesting features of the Pixel phones is that they include the built-in Google Assistant that allows for natural conversations with Google to find answers, explore search results, and perform a large array of tasks.

The Pixel and Pixel XL are available now for pre-order.

Nextcloud Announces Raspberry Pi-Powered Home Cloud Box

Nextcloud has teamed up with Canonical and WDLabs to launch Nextcloud Box, a secure, private, self-hosted cloud and Internet of Things (IoT) platform built around a Raspberry Pi. Nextcloud Box essentially comes with a case, cables, 1TB WDLabs hard drive, and a microSD card preloaded with Ubuntu Core and Nextcloud. The Raspberry Pi 2 is not included with the box; users need to buy it separately. The box is retailing for \$79 on the WD Store.

Jos Poortvliet of Nextcloud told us in an interview that the box is more or less a reference device that companies can use as a model for building and selling their own Nextcloud-based private cloud solutions. The box currently doesn't support Raspberry Pi 3, and it can't be accessed over the Internet. Poortvliet said that support for the RPi3 and remote access will be enabled in future software updates.

The most interesting feature of the Nextcloud Box is the Ubuntu Core operating system, which is designed for IoT devices and uses a transactional update model to keep the software on the device automatically updated.

WDLabs is an internal team within Western Digital that explores new ideas and possibilities.



Linus Torvalds Confirms Date of the First Linux Release

Linus Torvalds, the creator of the Linux kernel, has finally discovered the date of its first release: September 17, 1991. Twenty-five years later, to the day, Torvalds posted this message on his Google+ page:

Today is the 25th anniversary of the Linux-0.01 release, I do believe.

Normally the Linux anniversary is counted from the announcement email, which was August 25, because that was the actual public statement of intent. The 0.01 code drop happened a couple of weeks later and wasn't publicly announced.

Earlier during an interview with *Linux Pro* magazine, Torvalds said he was not sure about the actual Linux anniversary day, because there was possibly more than one date. Although August 25 is the date he announced the project he was working on, Torvalds said he never announced the first release publicly and simply uploaded it to an FTP server and emailed those who were interested in it.

Since then, he said, he lost all those email communications, so he didn't remember exactly when the first version of Linux was released. The only way to find out, he said, was to look at the timestamp on the first release. That's exactly what he seems to have done: The makefile of the first release is stamped September 17, 1991.

However, that doesn't change the official birthday of Linux. Torvalds settled that matter during a keynote discussion at LinuxCon North America (Toronto) when he said August 25 is the official date to use as a birthday.

MORE ONLINE

Linux Magazine

www.linux-magazine.com

Off the Beat • Bruce Byfield

Criticizing Less Than Free Hardware

With a boost from crowdfunding, efforts at free hardware are becoming a reality. Inevitably, though, most of these efforts are criticized as not being free enough, usually because of proprietary firmware.

The pi-top Revisited

Publication schedules mean that reviews are rarely the result of more than a few days of testing. Should the product develop long-term problems, they remain undocumented. Such is the case with the pi-top, the do-it-yourself laptop powered by the Raspberry Pi.

Paw Prints • Jon "maddog" Hall

True Love...and Microsoft Love

Over the past year I have heard with increasing volume about how much Microsoft loves Linux. However, I would like to substitute another word for "love," then tell you why I feel that way.

ADMIN HPC

<http://hpc.admin-magazine.com/>

NVDIMM and the Linux Kernel •

Markus Feilner and Johannes Thumshirn

Non-volatile dual in-line memory modules will provide storage as fast as RAM and keep its content through a reboot. The Linux kernel is already geared to handle the new technology and can even serve the modules up as block devices.

ADMIN Online

<http://www.admin-magazine.com/>

What's New in SQL Server 2016 • Thomas Joos

The focus in SQL Server 2016 is on mobility, cloud usage, and speed, with improvements to in-memory processing and security.

Technology Drivers for the IT industry • Udo Steinegger

The Internet of Everything, a fully networked and analyzed society, seeks to enhance the quality of life of all people and drive new technologies, products, services, and markets. But where are developments headed, and what are the driving factors, fields of application, and challenges?

It's Official: No Headphone Jacks in New iPhones

Apple has announced two new iPhones. The iPhone 7 features a 4.7-inch display, whereas the iPhone 7 Plus features a 5.5-inch display and comes with two 12MP wide-angle and telephoto cameras.

The most notable feature of the iPhone 7 family is the absence of a 3.5mm headphone jack. The only port on iPhone 7 is Apple's proprietary Lightning port. Apple is bundling a Lightning-to-3.5mm headphone jack adapter with each device so users can still use their existing headphones.

Users will not be able to use the wired headphone and charge the phone at the same time because the device only has a single Lightning port. That could be challenging to a lot of enterprise users who need to plug their phones into power outlets while making long phone calls.

Third-party vendors like Belkin are coming out with solutions like Lightning Audio + Charge RockStar, an adapter that comes with two Lightning ports so users can charge their phone while using wired headphones.



By killing the 3.5mm jack, Apple is moving to new wireless technologies for headphones. Although you can still use good old Bluetooth to pair wireless headphones, Apple has introduced AirPods, their new wireless headphones that use proprietary wireless technology to pair with iOS devices automatically. These AirPods are extremely smart and detect whether or not you are wearing them. Music will pause when you take them off and start again when you put them on. They also switch from stereo to mono when you take off one AirPods. AirPods' built-in microphone allows you to interact with Siri with a tap, without having to take your iPhone out of your pocket.

iPhone 7 and iPhone 7 Plus will be available in stores on September 16, 2016.

Image © nejron, 123RF.com

Opera's Password Sync Service Compromised

Opera Software has reported a security breach that compromised Opera Sync Service, the password manager for the Opera web browser. Opera has more than 350 million users and approximate 1.7 million people are using Opera Sync Service.

To its credit, Opera Software acted swiftly and notified its users of the breach by email. The company said in the blog post, "although we only store encrypted (for synchronized passwords) or hashed and salted (for authentication) passwords in this system, we have reset all the Opera Sync account passwords as a precaution."

Opera Sync users are also advised to change the passwords of third-party services, such as email, that they used on the browser, because this information might have been stored on the Opera cloud and therefore could have been compromised.

Password managers are used by web browsers to make it easy for users to store usernames, account information, and passwords on the cloud. All major browsers, including Google Chrome, Firefox, and Apple Safari offer such features. However, unlike Chrome and Safari, Opera Sync doesn't offer the additional security of two-factor authentication.



Image © jnphoto, 123RF.com

Zack's Kernel News

Chronicler Zack Brown reports on the latest news, views, dilemmas, and developments within the Linux kernel community.

By Zack Brown

Improving Kernel Locks

Waiman Long posted a patch to implement a new type of futex, called the throughput-optimized (TO) futex. He intended it specifically for use in cases where data throughput was more important than any other issue, including ensuring that all processes were given fair access to resources protected by that lock.

Other futexes used for similar purposes include wait-wake and priority inheritance (PI) futexes. On certain benchmarks, Waiman saw that TO futexes performed significantly better than the others.

Thomas Gleixner had some concerns. Primarily, he wasn't sure a new futex was really needed in the kernel. He also felt that user space would need to support a similar locking mechanism, possibly via `libc`, if the TO futex were adopted in the kernel.

Davidlohr Bueso replied, saying he felt that the TO futex was better than the alternative, and that if there wasn't a need for an additional futex in the kernel, the existing one should be the one to get rid of.

At one point Waiman explained part of his motivation for writing this patch. He said that at his company, he was "part of the team that help[s] large application vendors to tune their application performance on our large SMP systems. Those application vendors tend to use futex directly instead of relying on `libc`. We had seen spinlock contention in the futex could sometimes be a significant portion of the CPU cycles consumed depending on the workloads that were being run. We had been providing suggestions on the best practice of how to use futexes. But there is only so much

you can do with tuning their locking code implementation."

At one point, Thomas said he thought users would find it hard to know which futex to use under which circumstances. The differences between Waiman's TO futex and the other similar futexes were subtle and unintuitive. Thomas said, "So the benefit of these newfangled futexes is only there for extreme short critical sections and a gazillion of threads fighting for the same futex, right? I really wonder how the average programmer should pick the right flavour, not to talk about any useful decision for something like `libc` to pick the proper one."

Waiman replied, "Lock stealing will help performance when a gazillion of threads [fight] for the same futex. Optimistic spinning will help to reduce the lock transfer latency because the waiter isn't sleeping no matter the number of threads. One set of data that I haven't shown so far is that the performance delta between wait-wait and TO futexes actually increases as the critical section is lengthened. This is because for short critical section[s], the waiters of [a] wait-wake futex may not actually go to sleep because of the latency introduced by the code that has to be run before they do a final check to see if the futex value change[s] before going to sleep. The longer the critical section, the higher the chance that they actually sleep and hence their performance is getting worse relative to the TO futexes. For example, with the critical section of 50 pause instructions instead of 5, the performance gain is about 5X instead of about 1.6X in the latter case."

Waiman felt that the new TO futexes would perform better than other futexes most of the time, so maybe users wouldn't have to make the choice very often.

My personal sense is that kernel locking is one of those nightmare regions of code that make very little sense to the developers using them – and sometimes not even to the people writing the locking code itself. There's a reason why the Big Kernel Lock (BKL) lasted so long before being removed – it was far simpler and

easier to use than the odd array of locking options the kernel provides today.

But it's also good to pay attention to why the developers worked so hard to get rid of the BKL in the first place. Finer grained locking makes for a more efficient use of system resources, as well as a smoother user experience. Locks are taken and released many times per second on any running Linux system. Even a slight algorithmic improvement could add up to a huge speed improvement for the user. Maybe that's worth a little bewildering complexity.

Cleaning Up an Error Case

Al Viro noticed that the `writev()` system call had a non-intuitive behavior on a certain error condition. The `writev()` call is used to write an array of memory buffers to a specified file descriptor. However, when the address of one of the middle buffers was undefined, Al noticed that `writev()` would still write a single segment of data before giving up at the undefined address.

He felt it would make more sense to not write any data at all and to return the `EFAULT` error code. He consulted the POSIX standard and found that it was vague enough to allow the implementation he proposed.

Al gave an example of what he was talking about:

Suppose we have a buffer spanning 10 pages (amd64, so these are 4K ones) – 7 valid, 3 invalid,

```
vvvviiiiiv
```

and it starts 100 bytes into the first page. And write goes into a regular file on e.g.

ZACK BROWN

The Linux kernel mailing list comprises the core of Linux development activities. Traffic volumes are immense, often reaching 10,000 messages in a week, and keeping up to date with the entire scope of development is a virtually impossible task for one person. One of the few brave souls to take on this task is **Zack Brown**.

*tmpfs, starting at offset 31. We _can't_ write more than 4*4096-100 bytes, no matter what. It will be a short write. As the matter of fact, it will be even shorter than that – it will be 3*4096-31 bytes, up to the last pagecache boundary we can cover completely. That obviously depends upon the filesystem – not everything uses pagecache, for starters. However, the caller is *not* guaranteed that `write()` with an invalid page in the middle of a buffer would write everything up to the very beginning of the invalid page. A short write will happen, but the amount written might be up to [a] page size less than the actual length of valid part in the beginning of the buffer.*

Now, for `writev()` we could have invalid pages in any iovec; again, we obviously can't write anything past the first invalid page – we'll get either a short write or EFAULT (if nothing got written). That's fine; the question is what the caller can count upon `wrt` shortening.

*Again, we are *not* guaranteed writing up to [an] exact boundary. However, the current implementation will end up shortening no more than to the iovec boundary. I.e. if the first iovec contains only valid pages and there's an invalid one in the second iovec, the current implementation will write at least everything in the first iovec. That's `_not_` promised by POSIX or our man pages; moreover, I'm not sure if it's even true for each filesystem. And keeping that property is actually inconvenient – if we could discard it, we could make partial-copy -> `write_end()` calls a lot more infrequent.*

He said he would really like the logic of `writev()` to be, “if some addresses in the buffer(s) we are asked to write are invalid, the write will be shortened by up to a `PAGE_SIZE` from the first such invalid address,” which would bring the `writev()` behavior into line with `write()`'s behavior.”

Linus Torvalds replied, “I'm pretty sure you can and should do that.” He added, “there is no reason for the particular behavior.”

However, Alan Cox said that POSIX version 1003.1 said, “Each iovec entry specifies the base address and length of an area in memory from which data should be written. The `writev()` function shall always write a complete area before proceeding to the next.” This seemed to clearly contradict AI's preference. Alan remarked, “The moment you pass an invalid

address you are in the land of undefined behaviour, so I would read the standard as actually trying to deal with the behaviour in defined situations (e.g., out of disk space `mid-writev()`.”

Linus replied:

*But as you note, the EFAULT case is undefined behavior, so what that POSIX language is *really* about is presumably making sure that readers of a file cannot see the “later” writes without seeing the earlier ones.*

So you cannot do some fancy threaded thing where you do different iovec parts concurrently, because that could be seen by a reader (or more likely mmap) as doing the writes out of order.

Or, as you mention, the disk-full case.

The conversation ended there inconclusively, but for this sort of situation, I'd expect Linus to choose the sane behavior over POSIX compliance. He has always seemed to regard POSIX compliance as a convenient guide rather than as something to be blindly adhered to, and in this case, he seemed to want to find reasons to bring the `writev()` behavior into consistency with `write()`.

Abortive Attempt at Alerts from User Space

Luis R. Rodriguez noticed a possible race condition at bootup, if the user used the `kernel_read_file_from_path()` call to read a file from the system's filesystem. If the call occurred before a certain point in the boot process, the filesystem wouldn't be available and the read would fail; however, after a certain point, the read would succeed.

Luis described his patch, saying, “We define kernel critical filesystems as filesystems which the kernel needs for `kernel_read_file_from_path()`. Since only user space can know when kernel critical filesystems are mounted and ready, let user space notify the kernel of this, and enable a new kernel configuration which lets the kernel wait for this event before enabling reads from `kernel_read_file_from_path()`. A default timeout of 10s is used for now.”

Linus Torvalds took one look at this and said, “I really think this is a horrible hack. It's basically the kernel giving up, and relying on user space to give a single flag, and it's broken nasty crap. Worse, it's broken nasty crap with a user interface, so we'll be stuck with it forever. Please no.”

He asked which drivers needed this and suggested that they simply be fixed to not hit that particular race.

Dmitry Torokhov agreed that the user-space interaction was bad, but he felt it was necessary nevertheless. He didn't see how the drivers could possibly be “fixed” to avoid the problem. He said:

Some devices do need to have firmware loaded so we know their capabilities, so we really can't push the firmware loading into 'open'. If your touch controller for some reason decided to crap over it's nvram and comes in bootloader mode it is nice for it to slurp in config data or firmware so use does not have to trigger update manually. And while the controller is in bootloader mode we can't create [an] input device because we do not know what capabilities to declare.

These devices we want to probe asynchronously and simply tell [the] firmware loader to wait for firmware to become available. The problem [is] we do not know when to give up, since we do not know where the firmware might be. But user space knows and can tell us.

Meanwhile Bjorn Andersson said that he had specific cases that would be helped by Luis' code. He said, “I have several cases where `remoteproc` drivers are used [to] boot DSPs upon boot of the device, but the firmware files are way too big for being stored in `initramfs` and all consumers of the provided services are (semi-) probable as the remote processor is booted. I.e., we need some way to figure out when these files become available so we can bring these remote processors up.”

One solution that Linus had offered earlier was to require drivers running into this problem, to be built as loadable modules. But Bjorn remarked, “I really do not like the deployment issues that come with kernel modules during development. (The firmware and `remoteproc` driver normally do not have the same flow through a development process.)”

So while he agreed that Luis' proposal was “a horrible hack,” he said, “I would appreciate a automagical mechanism that would relieve user space from having to signal to the kernel that the firmware partition has been mounted.”

Linus wasn't satisfied with any of these arguments. He said if a driver definitely needed firmware (as opposed to it being simply optional), then “why don't we just tie the firmware and module together?” He

added, “Really. If the driver doesn’t work without the firmware, then why the hell is it separated from it in the first place?”

Bjorn responded from his own personal experience:

... generally there’s a much stronger bond between the kernel and the driver than between the driver and the firmware in my cases.

E.g., we have a single `remoteproc` driver loading and controlling the Hexagon DSP found in several Qualcomm platforms, so a single kernel binary could (practically) load hundreds of variants of the firmware.

Both the kernel binary and the firmware in this example are side-loaded onto the device during development – independently of each other, as they are developed by different teams (or maybe even different companies).

I assume that you’re not suggesting to actually tie the module together, as that would be practically difficult and a waste of resources.

Which leaves us with the suggestion that we should store the kernel module with the firmware file, which is just infeasible from a few practical reasons – again mostly related to the development flow and how the files are contained on the devices.

Bjorn reiterated that Luis’s proposal seemed like an ugly hack – but he just didn’t see an alternative.

Linus said, “all these arguments make no sense at all,” and added, “Let me be very clear. I’m not applying that shit-for-brains stupid patch, and will not be pulling it unless somebody tricks me into it.”

He went on:

If the driver doesn’t work without the firmware, then anybody who distributes the driver binary without a firmware is just **fundamentally** doing something insane. You may do it for **development** purposes, but doing so for actual **use** would be entirely pointless.

See my point? If a distribution is distributing the driver without the firmware, then what the hell is the point of such a thing?

But even if you decide to do that for some odd reason, the patch is still just stupid. Instead of adding some crazy infrastructure for “now I’ve mounted everything,” you could equally well just:

- (a) Make the driver fail the module load if it cannot find a firmware binary
- (b) After user space has mounted everything, just do “`insmod -a`” again (or `insmod` just that driver).

See? The point is, this “generic” hacky interface is just stupid. It’s not adding any value. If you add user space “I’m ready now” points anyway, you might as well make those points do the right thing and just load the module that is now loadable.

We could mark such “late loading” modules explicitly if people want to, so that you can automate the whole thing about delaying the loading in user space.

At no point does it make sense to say “I have now mounted all the important filesystems.” Maybe the firmware is extracted later by user space downloading it from the Internet, and the module will then work only after that point.

This whole “I have mounted important filesystems” is just pure and utter garbage. Stop pushing this shit.

Luis replied that, “this isn’t just about firmware since we now have a generic API to read files from the kernel, so `kernel_read_file_from_path()`. Firmware is now just **one** user case. Also a complexity here is this is not just for modules but also for built-in drivers as well. And you want the option to update the files without the driver. The proposed solution provides a generic broad syfs entry for letting user space inform the kernel when files from the filesystems where it would typically read from (I’m calling them critical filesystems for lack of a better term) can be accessible. Something more specific requires a bit more thought given [that] this is not anymore about just firmware, [but] must also address built-in drivers, and allow for updates.”

He added, “I can see the syfs approach being considered hacky – but I gladly welcome an actual alternative suggestion.”

Luis also pointed out to Linus, that “the only reason I proposed this was to get the ball rolling in terms of finding a solution to the problem for why some folks claim they **need** the firmware usermode helper. Granted, upstream we only have two explicit users left. I’m told some out-of-tree users still need and use the usermode helper. They claim that without it there is the race between being ready and driver asking for the firmware. I was told there were quite a bit of out-of-tree hacks to address this without using the usermode helper, the goal of this patch was to create the discussion needed to [find] a proper resolution to this.”

Several folks jumped into the conversation at this point, offering suggestions

for possible alternative solutions. But Linus was having none of it. Finally he offered some back-handed advice for how to go about these sorts of kernel feature design discussions. He said:

The reason I’ve hated this whole discussion is that it’s full of ‘let’s re-architect everything’, and then it has these horribly warty interfaces. It’s classic second-system syndrome.

Just do **one** thing, and do it well. Don’t change anything else. Don’t force existing drivers to use new interfaces. Don’t over-architect, and don’t do stupid interfaces.

If user space mounts a new filesystem (or just unpacks files from a tar file that has firmware images in it, for Chrissake), that is not some magical “critical mount event.” The whole concept is just stupid. Is it a “mount event” when the user downloads a new firmware image from the Internet?

HELL NO.

But what is equally stupid is to then dismiss simple models because [of] some totally unrelated ‘beyond firmware’ issue.

Anything that is ‘beyond firmware’ shouldn’t even be discussed, for Chrissake! It has nothing what-so-ever to do with firmware loading. If there ends up being some common helper functions and shared code, that **still** doesn’t make it so.

Basic rules of thumb:

- (a) Don’t over-design.
- (b) Don’t have stupid illogical interfaces.
- (c) Don’t conflate different issues just because you think they may have shared code.
- (4) Be consistent. Don’t make up new interfaces, and most certainly do **NOT** dismiss something just because it’s what we have done before.

The discussion petered out shortly afterward. This was an example of Linus really slamming some developers for using practices that he didn’t like, but it was also an example of developers bearing up under that kind of blistering criticism, avoiding counterattacks, and simply proceeding with a consideration of what to try next. This is actually the usual case when Linus comes down hard on a particular approach. No matter how harsh he becomes, the developers nearly always simply extract the technical considerations and respond to them as best as they can. ■■■



An interview with Greg Kroah-Hartman

World Domination

By Swapnil Bhartiya

Greg Kroah-Hartman is one of the friendliest faces of the Linux kernel community. He maintains the stable branch of the Linux kernel and participates in many more Linux projects. He works at the Linux Foundation as a fellow, a job that allows him to dedicate his time to the Linux kernel. We meet each other at almost every Linux-related conference. I sat down with Greg to talk about the kernel, 25 years of Linux, and what he does with the rest of his life.

Linux Magazine: What and how much has Linux achieved in these 25 years?

Greg Kroah-Hartman: We had this joke: “Total world domination was our goal.” Everybody laughed at us. Then we did it, and nobody even noticed. I mean, we’re everywhere. We’re going places that Linux has never been. It’s crazy to think that this project we all started working on for fun, myself included, has done this. It’s insane.

I was talking to people the other night, and we were discussing that any chip that has over four megabytes of memory runs Linux. We’re even trying to get it on smaller chips now. We are everywhere – from giant supercomputers to devices with less than 4MB of RAM.

LM: Linus started Linux in 1991, when did you get involved with it?

GKH: I started using Linux in the 1990s when Oracle first got ported to it because I was working for a company that ran Oracle on SCO machines. That was the first time I ran Linux. I used it a little bit and had fun building my own kernel, but I never saw anything that I could contribute.

In 1995, I was working for a barcode scanner company called PSC Inc. doing

embedded software development. I wrote firmware for a barcode scanner that used USB. Linux didn’t have a USB stack then. I had to test my USB scanner on all operating systems to make sure I got my USB code right. I saw Linux and watched the USB stack being developed for it. Linus and other developers started writing a little bit of code, and USB support started to get implemented. With that codebase, I saw little places where I could contribute, fixing odd bugs and adding small features. Linus took my patches and it was exciting. I said: “This is fun.”

LM: Linus started Linux to scratch his itch, but you got involved due to your day job, or was there any personal itch to scratch?

GKH: It was part of my job, but I had my own itch. I had a USB device that didn’t work on Linux. My wife went away on vacation for a weekend and said: “Why don’t you write a driver? You always talk about that.” I said OK, I’ll write a driver. I wrote a driver over the weekend and submitted it, and I swear within an hour people came back pointing out problems telling me: This is wrong; this is wrong; this is wrong.

It felt awesome. They were critiquing my code, and I was learning from it. I always want to keep learning, so I said “Yes, you are right. This is wrong, this is wrong, and this is wrong.” I iterated and fixed problems with it. It got accepted into the kernel. To me, it was fun.

I think feedback is very important. That feedback loop of people pointing out errors or problems with what you’re doing is a very traditional, I guess scientific, method. And I love it. That’s how we get better.

LM: At that time, you were working at a proprietary company. How different

was the development inside the company as compared to open source? What kind of feedback would you get when you wrote code internally?

GKH: PSC Inc. was selling barcode scanners. We had a traditional model of software development, which you followed to write the code. I would get a lot of review about the specs, but when you wrote the code, there was nobody to review the code. It would go to testers and they would find problems. There would be a feedback loop in that model, but there was a gap. You write all this code, but you don’t get any feedback at early stages as other developers are also writing their own stuff, and then all of it goes to testers. You didn’t get feedback from fellow developers.

Anybody who has written code knows that it’s better to find problems as early as possible. And that’s the way Linux works. We find most bugs and problems before anybody actually runs the code. The code gets reviewed at a very early stage.

People send me patches. I was flying across the Atlantic for this conference, and I had 800 emails about just one subsystem. I went through them and found out that 200 of them applied, so I rejected the rest. Well, that’s a bad percentage. That’s fine. The point is that the review of the code was done immediately, at a very early stage. All of that was just from code review, before the code was even run. We go back and fix things immediately. That helps to catch things really early in the process, and that’s good. It turns out to be a much better way of developing software.

LM: So how did you get from PSC to the Linux Foundation?

GKH: I wrote that driver a long time ago, and we didn’t use Linux or anything;

we were using embedded systems. Then things took off in 2000, around the dot-com boom. Suddenly there were all these new jobs. That was also the time when I was getting all these emails from people who were using the code that I had written. That's when I realized more people were using the stuff I wrote for free as compared to anything I ever got paid for (chuckles).

Even though the company was relatively successful – we sold barcode scanners that ran everywhere – those emails were exciting. I realized that it was so much fun. Then I got offered a job that essentially said: “You want to do this full time?” I was like “Yes!”

After PSC, I worked for WireX for a while before moving to IBM. I worked at IBM for a number of years and created the driver model with Pat Mochel, who was working at OSDL. Then I was on the IBM kernel team. After that, I moved to SUSE, and now I am a fellow of the Linux Foundation.

LM: What is a fellow of the Linux Foundation? What's your actual job?

GKH: My official job title is: “Linux Foundation can't tell me what to do, and I can't tell them what to do.” (Chuckles) I do the stable kernel releases. I'm paid to be able to come and do my development without having to create a product or anything else like that, so it's fun.

I go around and help companies, teach them how to get involved in kernel development. Lately, I have been working with other projects to teach them how to scale. For example, Kubernetes hit a scaling wall recently with the number of contributors. I showed them how the kernel team solves these problems and helped them in figuring out that maybe they can change their development model and do better. A few years ago, I talked to Docker developers on how they can scale better. I have done that for some other companies and groups, like the Open Platform for Network Functions Virtualization (OPNFV) project.

LM: Has the development model changed over the years, or is it still the same?

GKH: The development model has changed a lot. We have changed it over time. We started Kernel Summit so we can meet each other and figure out how to continue to do this better. Part of our Kernel Summit is to talk about what is

going wrong and what we need to do to fix it.

At one of the kernel summits, Larry McVoy showed us how to use BitKeeper, the distributed source-code control system. He showed us how we could distribute the source code and distribute development in a way that sped up the kernel development immensely.

Previously, I'd send an email to Linus with patches, he'd apply them to his local tree, and every week he would do a release. But you are left wondering whether he took your patch or not; you had to figure it out on your own. BitKeeper changed that. If I sent him patches, he'd push it out to his BitKeeper public tree, and I could see that.

Now it's even better with Git, because it's all distributed and scaled. I can work on the plane, without any email or Internet access. When I land, I push publicly to my *kernel.org* tree. The test infrastructure kicks in, and if the code passes all the tests, it moves to the next stage.

Now we have automatic testing and a distributed development model, which is so much faster, because we are dealing with lots and lots more people. Fifteen years ago, we were doing two patches an hour. At that time, we all thought that was insane; it was faster than anybody else. Now we're at eight patches an hour, and every year it keeps going up. The number of developers is increasing every release. The number of companies contributing is increasing. It's just crazy.

LM: You are often seen as second in charge after Linus. What real challenges do you see ahead for the Linux project?

GKH: I'm not second in charge. Linus and I have different roles. We deal with different people. Every year I keep saying there is no way we're going to scale faster if we're going to get bigger. We passed the other operating systems' development size and speed years and years ago. Now even Microsoft runs Linux. It's amazing. My worry is that the only thing that's going to hurt Linux is Linux itself. If we mess up. My goal is not to mess us up, and I try to help see the problems. We continue to talk about issues at the Kernel Summit and try to find ways to fix them. We are constantly paying attention, so that we are able to say that, yes, we're doing something stupid here, let's fix it. It's an iterative process.

LM: While we are talking about challenges, I recall Linus once said that Linux can't shrink anymore, and if you want to work on really small devices, you will have to end up with other alternatives. However I have talked to some companies that say you can still strip it down to do whatever you need...

GKH: I've been working a lot with those people that do that. You can run Linux on 4MB; it's a piece of cake. And you can get it down to 2MB. Marcel Hoffmann of Intel delivered a talk at LinuxCon on how to get it even smaller. The other night we were talking over beer (and I always get in trouble when I do that), and I said we can figure out a way and probably get it down to 512K, maybe even smaller. It really depends on what you want to use it for. Hoffmann, for example, is a Bluetooth guy who wants to make an IoT device. He wanted a Bluetooth stack, and we figured out a way to do it. But it's hard to even find chips that don't have four megabytes of RAM.

LM: The reason I asked that question was Google's new operating system, Fuchsia, which is targeted at really small devices. They have not made any public announcement, but it's on GitHub. Have you looked at it?

GKH: Yes, I have seen it, and I know those guys. If you look at the code, it's awesome.

LM: But they are not using the Linux kernel?

GKH: No, they're using Little Kernel. It's a tiny operating system that has been around for many years. It's on most Android phones; it's your boot loader. So what they did was they took Little Kernel and made it usable. It's gotten bigger now and has SMP x86 support. But it doesn't look like a normal operating system; it doesn't have a userspace component. Fuchsia works on adding userspace support to Little Kernel. Now you can run processes and do other fun things like running programs in Dart. Little Kernel is written by the same guys who wrote the original Android code, BeOS, and Palm. They're having fun making another operating system.

The good thing about Google is that they release thousands of different GitHub projects. This is just one of them, and it's been public for a long time. Google open sources a lot of stuff.

LM: How's your work/life balance? Do you work from home?

GKH: Yeah, I've worked from home ever since I started working for SUSE. Work/life balance is tricky. In the beginning I was really strict. I would say: "Okay, I'm working this hour to this hour and then that's it." But over time, I will be talking to my wife and ask, "What are you doing? Let's go have a coffee." Then you realize you can still get all your work done.

Since I work from home, I worry about my children, because I saw my father always go off to work, and he had very good work ethics. When my son was really little he would say: "No, my dad doesn't have a job; he just does emails in his pajamas."

LM: So Linus works from his bathrobe and you wear pajamas?

GKH: Yes, I work in pajamas. In my previous house, I had a basement office. We recently moved to Paris, and I have carved out an office space with a desk in our apartment.

My kids are growing up. My daughter has given talks at LinuxCon. She has worked LinuxCon events. She knows what I do and how this works. My son was at LinuxCon Japan last year, so they know what's going on and they see that it's a very different type of job.

I work from home, and I love it; it's great. But I travel a lot, too.

LM: So when you travel around the globe, talking to these companies, what's your goal?

GKH: My goal is: I want these companies to become part of the kernel community, because everybody contributes to Linux in a selfish way. They want Linux to work in a selfish manner. And that's fine, because when you contribute to Linux in a selfish manner, it benefits everybody.

Let me give you an example. It was the embedded guys who wanted good power management in Linux to save batteries. We got it to work, and now giant companies running giant supercomputers are saving billions of dollars thanks to the power management of Linux. It benefits everybody that somebody was being selfish.

LM: These days we hear a lot about security. Are we introducing more bugs due to Linux's popularity and its rising share of media attention?

GKH: I think the security issues have always been there. My very first job in Linux was to work on a secure version of Linux at WireX. The person in charge of that, Crispin Cowan, wrote some tools and modified gcc to help prevent stack overflows. We had a DARPA contract to do research on how to make things more secure, and we wrote some USENIX papers based on those results. That was fifteen years ago; we were already talking about how security needs to be done better; what we need is to be more proactive.

It's just that now people are finally realizing that, yes, this is a problem. It's always been there, and we need to pay more attention to it. We need to do better things. We need to avoid being reactionary. We need to do things in defensive ways, like adding code to the compilers to check for stack overflows automatically. Or look at all the GR security work that's happened and get that in.

Kees Cook from Google, and Konstantin Ryabitsev, the *kernel.org* maintainer, look at it like cars and airbags. You can make these big steel death machines that go really fast but you need to protect people inside them, so you need to be defensive. Now cars have crumple zones, airbags, and all these other safety things that aren't there to make the car go faster, but to protect the people. As kernel developers, we have to accept that we need to take the overhead of protecting things better and create crumple zones and airbags.

LM: In a previous conversation, you talked about how the kernel community is doing all this work, but the changes don't reach target devices, because vendors are not making it happen. Google is trying to change how Android devices are updated. What's needed, a cultural change or technological change?

GKH: It's totally a cultural change. It's the old model of "Hey if it works, don't touch it. It's not going to break."

Well, the world changes. If you look at the old version of Linux, it worked great back then based on the ideas of what we knew then, but the world changes. You never want to put an old version of Windows on the Internet now, and you wouldn't want to put an old version of Linux on the Internet now either because we know all the problems.

You have to be able to update. It's a cultural change to tell people that it's

okay to accept changes. 12-13 years ago, the kernel community said we should make the guarantee that we will not break your system. We give you the safety of knowing that, if you upgrade, you're okay, and we have backed that up. We've proven we can do that.

It's overcoming that fear. People will accept change, but they will accept the change that they want; they don't want the other changes. They're worried about those changes breaking something.

To make such changes acceptable, you have to put an infrastructure in place to test the changes. And companies are doing a lot of work around that. Google is working on updating Android faster. They're working on infrastructure to make it easier for SoC vendors to do the testing so they can rely on it. They can test their machine, and they can put a new kernel in there and see if passes all the tests. If everything looks good, they can push it out to the world.

LM: Where do all these new kernel developers come from?

GKH: Over the years, you'll see we had people contribute to Linux, and then they go work for a company. Then they become a manager, then they become a vice president, and then they become a CTO, and they're still Linux developers.

One of the Japanese contributors worked for a company, and he wanted to show his engineers that anybody can contribute. He was a vice president, and he contributed documentation updates. Then he went to his engineers and said "I can do this, why can't you?" That's how Linux has succeeded.

Then people move around. The funny thing about the kernel community is that we all change jobs, but we all still do the same thing. It just keeps growing. It creates communities, people teach other people, and we get more people involved.

Efforts like Google Summer of Code and Outreachy get more and more people involved. The Eastern Europe Outreach program has grown so huge that there's a whole university pumping out kernel developers now. There's a class taught at the university where one of the projects is to contribute a code patch to the kernel.

These people are coming out of college with Linux Kernel experience, and then they're going off into other companies. It's been amazing. It's just growing that way. ■■■



ENIGMA

MORE TO DECIPHER

Enigma: A USENIX conference on security and privacy in context

Join a diverse mix of experts and enthusiasts from industry, academia, and government for three days of presentations and open sharing of ideas.

Our focus is on current and emerging threats and defenses in the growing intersection of society and technology. Our goal is to foster an intelligent and informed conversation with the community and with the world, including a wide variety of perspectives, backgrounds, and experiences.

The full program and registration are available now.

enigma.usenix.org

JAN 30–FEB 1 2017
OAKLAND, CALIFORNIA, USA





Embedding geotags in digital photos

X Marks the Spot

Add location data to your best digital images with **digiKam** and **GPS Correlate**. *By Karsten Günther*

Many image formats let you store the GPS coordinates for the image location as metadata. Storing location data with the image makes it easy to document where the image was taken. Some mapping tools will even let you plot the geotagged images on a map; you can click an icon on the map and see what the view was like from that location.

Most smartphones, which typically have built-in GPS capabilities, automatically add location data to captured images. But conventional digital cameras, even many premium cameras, don't have a way to sense and record GPS data.

The good news for users of these cameras is that Linux has some tools that will let you add location data to digital images after the pictures are taken. Most of these tools work with a GPX track. A GPX track is a log of coordinates from a recent hike or road trip. Turn on GPS tracking on your smartphone or GPS device, then start walking or riding. The device will record the data as a track that you can then display using a mapping tool.

This article describes a pair of tools for geotagging digital images: **digiKam** and **GPS Correlate**.

Converting Your Track

For many years, most GPS devices supported GPX format out of the box; however, some recent devices now use other formats. For instance, today's Garmin devices use the FIT (Flexible and Interoperable Data Transfer) format, which you will need to convert to GPX to use it with the geotagging tools described in this article. You can use a tool called **GPSTabel** [1] to convert virtually any track format.

The **GPSTabel** program supports almost all relevant formats and lets you convert relatively easily. To convert, for example, a FIT file, use the `garmin_fit` option as a parameter for the **GPSTabel** program (Listing 1).

If you favor a GUI, the **GPSTabel** tool does have a graphical interface named **Gebabel**, but my impression is that **Gebabel** doesn't really make using **GPSTabel** much easier, because it also ultimately requires input at the command line.

LISTING 1: Converting a FIT

```
$ gpsbabel -i garmin_fit -f <filename> -o gpx -F <filename>
```





Once you have the GPX tracks for the tour, you can start tagging. The coordinates of each track must exist in its own file. You can collect all your tracks in a directory and specify all the files to correlate. The programs ignore tracks that you recorded on days other than the days the images were processed.

digiKam

If you already use the open source digiKam photo tool [2] to manage images, you might already be familiar with a feature called *Edit Coordinates*, or something similar – the name has changed several times in recent versions. You will find it under *Image | Geolocation | Edit Coordinates*.

When you select *Edit Coordinates*, digiKam displays a wizard to let you associate your photos with locations on the GPX track. You can either position the images on the map with your mouse to associate the image with the correspond-

ing coordinates, or you can use using the *GPS Correlator* tab on the right margin (Figure 1).

If you load multiple GPS tracks at the same time, digiKam shows them in different colors to distinguish between tracks.

The *Manual* option and *Fine offset (mm:ss)* below *Camera time zone* help you synchronize the clocks in the camera and your GPS device (Figure 2). *Interpolate* tells the application to calculate any missing timestamps automatically. The synchronization feature could cause problems if you connect multiple tracks over a period of time: synchronization is only intended to plug minor gaps in the tracks, such as when woods or buildings prevented reception.

Pressing *Correlate* accepts the data in the images. To check whether the correlation was successful, check randomly selected images to see whether they were positioned correctly on the map. If you have defined an offset, check that the sign is correct. If you receive an error message, look for non-correlated images.

The easiest way of doing this is to open the images as a list and then sort them by latitude. Images that the application did not map are at the top of the list. Try to find out why they were not matched: Missing a track? Faulty Exif data? If you want to position the images manually, now is the right time to do it.

You can stop the action by pressing the *Apply* button, which tells the program to write the data to the images. If you want to write data to RAW files, you will need to enable the option for writing metadata in the digiKam configuration. Choose the *Settings* menu below *Configure digiKam*, select the *Metadata* tab, and enable the option labeled *If possible, write metadata to raw files (experimental)*. Alternatively, you could add the metadata to XMP sidecar files.

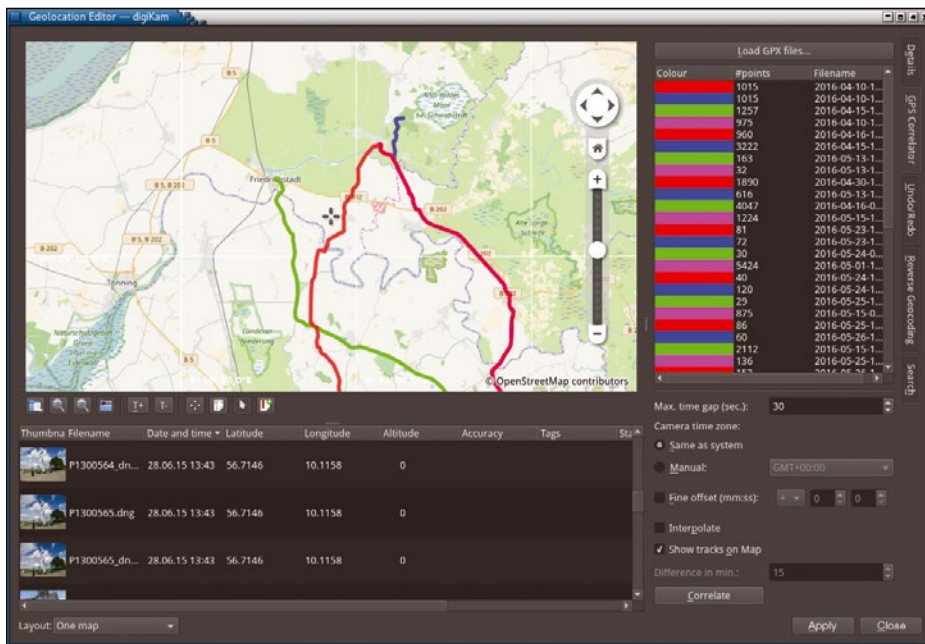


Figure 1: Using digiKam, users can transfer coordinates from the GPX tracks to images.

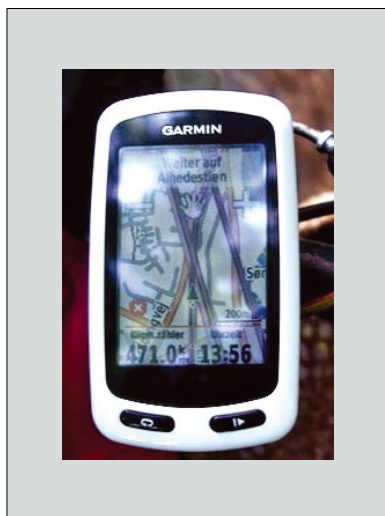


Figure 2: If necessary, adjust the navigation device's time with the time the camera stored in the Exif tags.

GPS Photo Correlate 1.6.2.git						
	File	Latitude	Longitude	Elevation	Time	State
1. Add Photos	P1390508.dng	55.799333 (N)	9.373422 (E)	86.60m	2016-06-23 11:00:43	GPS Data Present
	P1390507.dng	55.758744 (N)	9.419986 (E)	110.40m	2016-06-23 10:34:45	GPS Data Present
2. GPS Data	P1390509.dng	55.799755 (N)	9.372447 (E)	88.00m	2016-06-23 11:01:00	GPS Data Present
	P1390510.dng	55.801955 (N)	9.372275 (E)	87.80m	2016-06-23 11:02:20	GPS Data Present
3. Set options	P1390511.dng	55.810306 (N)	9.368657 (E)	85.60m	2016-06-23 11:05:40	GPS Data Present
	P1390516.dng	55.852433 (N)	9.363350 (E)	114.40m	2016-06-23 11:29:09	GPS Data Present
	P1390524.dng	55.865334 (N)	9.364916 (E)	118.40m	2016-06-23 11:37:59	GPS Data Present
	P1390525.dng	55.867572 (N)	9.365278 (E)	119.60m	2016-06-23 11:38:56	GPS Data Present
	P1390526.dng	55.885994 (N)	9.377433 (E)	114.20m	2016-06-23 11:47:51	GPS Data Present
	P1390527.dng	55.886103 (N)	9.377722 (E)	113.20m	2016-06-23 11:47:52	GPS Data Present
	P1390528.dng	55.910699 (N)	9.400072 (E)	73.00m	2016-06-23 11:59:28	GPS Data Present
	P1390529.dng	55.910699 (N)	9.400072 (E)	73.00m	2016-06-23 11:59:30	GPS Data Present
	P1390534.dng	55.968082 (N)	9.395968 (E)	114.60m	2016-06-23 12:41:11	GPS Data Present
	P1390535.dng	55.968312 (N)	9.395857 (E)	114.80m	2016-06-23 12:41:18	GPS Data Present
4. Correlate!	P1390536.dng				2016-06-23 12:53:44	Ready
Other Tools	P1390537.dng	56.006126 (N)	9.419551 (E)	97.80m	2016-06-23 13:08:22	GPS Data Present
	P1390538.dng	56.006126 (N)	9.419551 (E)	97.80m	2016-06-23 13:08:23	GPS Data Present
	P1390539.dng	56.011704 (N)	9.430101 (E)	88.00m	2016-06-23 13:13:20	GPS Data Present
	P1390541.dng	56.013929 (N)	9.432238 (E)		2016-06-23 13:28:16	GPS Data Present
	P1390540.dng	56.013236 (N)	9.432717 (E)	76.60m	2016-06-23 13:14:18	GPS Data Present
	P1390542.dng	56.014432 (N)	9.431579 (E)		2016-06-23 13:28:18	GPS Data Present
	P1390543.dng	56.014839 (N)	9.431544 (E)		2016-06-23 13:28:21	GPS Data Present
	P1390544.dng	56.013279 (N)	9.432467 (E)	79.80m	2016-06-23 13:34:26	GPS Data Present
	P1390545.dng	56.015435 (N)	9.433899 (E)	91.20m	2016-06-23 13:38:02	GPS Data Present
	P1390547.dng	56.017758 (N)	9.434856 (E)	91.00m	2016-06-23 13:39:14	GPS Data Present
	P1390546.dng	56.016158 (N)	9.434163 (E)	90.80m	2016-06-23 13:38:22	GPS Data Present

Figure 3: GPS Correlate's graphical interface provides a simple alternative to the digiKam wizard for geotagging photos.



TABLE 1: GPS Correlate Options

Option	Function
-g <datafile>	Specify GPX track used
-z <+/-HH[:MM]>	Time offset
-i	Disable (linear) interpolation
-n	Stop writing to the file
-m <seconds>	Maximum time for points outside of the tracks
-s	Show stored geographic information from the files
-o	Output in machine-readable format
-r	Remove geo-information from files
-t	Interpolate between the track segments
-M	Do not change modification time of file on writing
-f	Repair broken time stamps
-O <seconds>	Offset time for images
-v, --verbose	Output more detail in messages

You manage the command-line variant through a series of options. Table 1 summarizes the most important. By default, the tool changes the image files it processes, even RAW files, without providing feedback. The *Don't write* option in the GUI and command-line parameter `n` prevent this automatic write option.

Follow-Up

Now that you have tagged the images with geo-information, what can you do with it? digiKam offers two ways to access this information. First, you can view a map with the figure locations shown on it. To see the map with the associated images, select the *Geolocation* tab (Figure 4).

GPS Correlate

Another option for geotagging digital images is GPS Correlate [3]. GPS Correlate consists of two components: a program for the command line (`gpscorrelate`) and a GUI (Figure 3). Like digiKam, GPS Correlate can work with RAW files as well as JPEGs.

The GPS Correlate interface is simpler than digiKam's wizard. The tool displays the data very clearly, which really helps if you have to manage many images. However, some stumbling blocks spoil the overall good impression: If the timestamps were not recorded precisely, the software is significantly less capable of automatically correlating images than the digiKam wizard. Also, the software does not let you automatically sort the non-mapped images so that entries appear within a block in the list. Finally, a preview feature, which is always useful for troubleshooting, is missing.

Each image appears on the map, which makes it very easy to give a presentation or informal talk on the hike.

The other option is a map search, which you can select in the left sidebar. The map search window is similar to the view in Figure 4, but it has some important additional features. Below *Search in area*, you will find three buttons that let you define the search area; digiKam only displays images from this area in the main window.

The first button lets you drag a frame to define the area. The second lets you define the search with a sample image; the third deletes existing areas again.

In addition, and for faster access to the most frequently used images, you can save individual search results. Enter a name for your search in the box below *Show entries without geolocation information*. Now, the application generates a virtual album with the results.

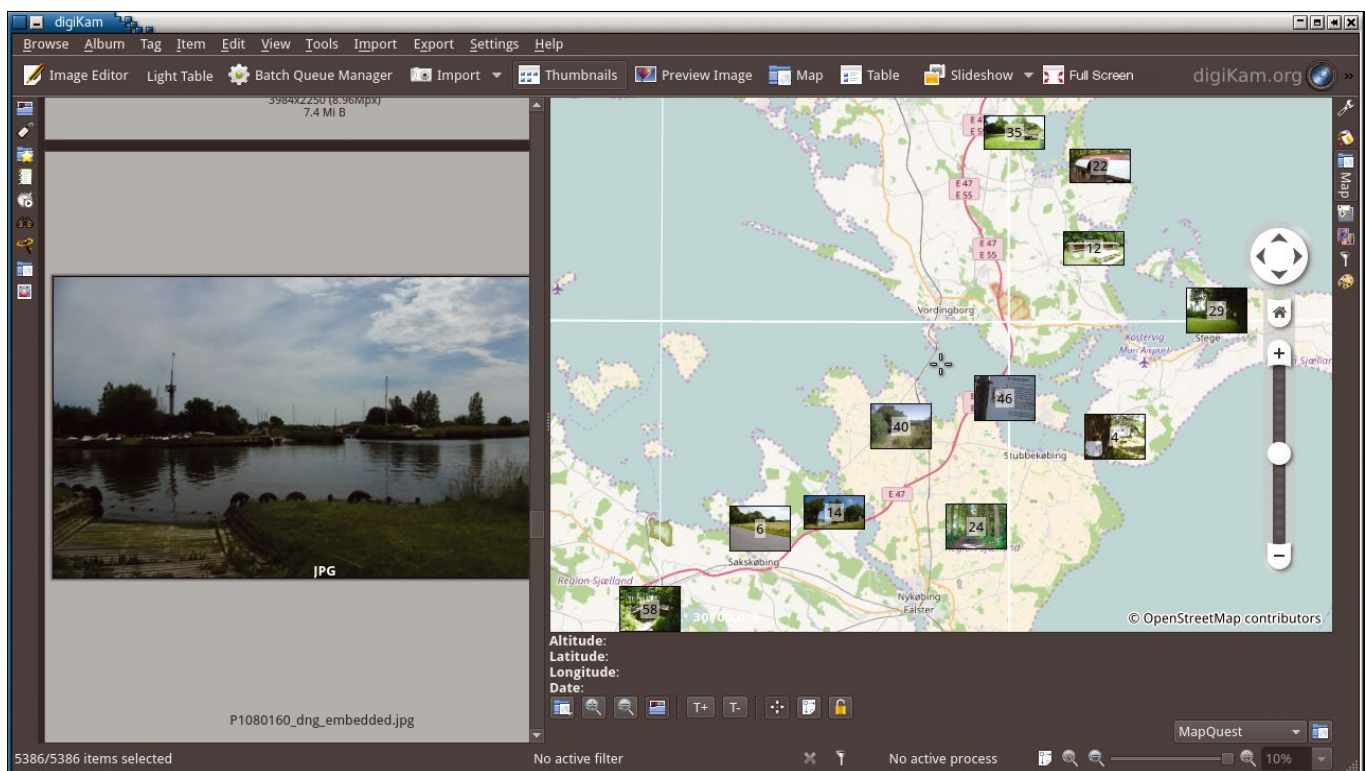


Figure 4: The digiKam photo manager lets you display images on a map.

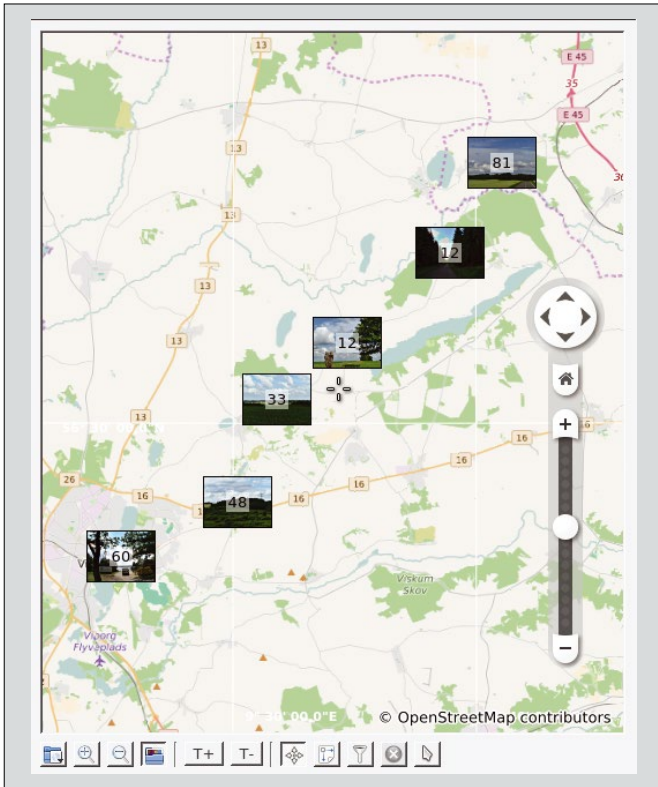


Figure 5: The map views in digiKam offer many options, not all of which are meaningful and useful.

Maps

The windows used by digiKam for the maps are powerful, but the layout is slightly complicated. digiKam offers some options for configuring the display. Select the sources on the left to define what view form you want to use. The default variant *Marble*

(*virtual globe*) is often a sensible choice – *Atlas* is not suited for locating images because the resolution and level of detail is far too low. You will do far better with *OpenStreetMap* (Figure 5). The next two buttons control the scale of the displayed map, much like the slider in the preview. Pressing *T+* and *T-* lets you set the size of the icons displayed for the images. If you set too large a value, digiKam groups the icons, which obscures the exact positions.

GPX Viewer

GPX Viewer [4] is a handy option to load and view a GPX track. The application also displays an elevation profile of the track (Figure 6). The trick lies in the *Playback* function, which is hidden away in the *Detailed track information* tab.

The buttons let you switch the track off once again on the map. Given a suitable magnification level, you can immediately see where you were at a given time, see the breaks in the track, and find the dates and places you took the pictures.

Conclusions

With relatively little effort, you can use GPX tracks to geolocate images. For most users, digiKam is the best choice because it uses a simple approach and has a graphical wizard. GPS Correlate is a viable alternative for the command line. You will find GPS Correlate, as well as GPX Viewer, in the repositories of all popular distributions. ■■■

INFO

- [1] GPSBabel: <https://www.gpsbabel.org>
- [2] digiKam: <http://www.digikam.org>
- [3] GPS Correlate: <https://github.com/freefoote/gpscorrelate>
- [4] GPX Viewer: <http://blog.sarine.nl/gpx-viewer/>

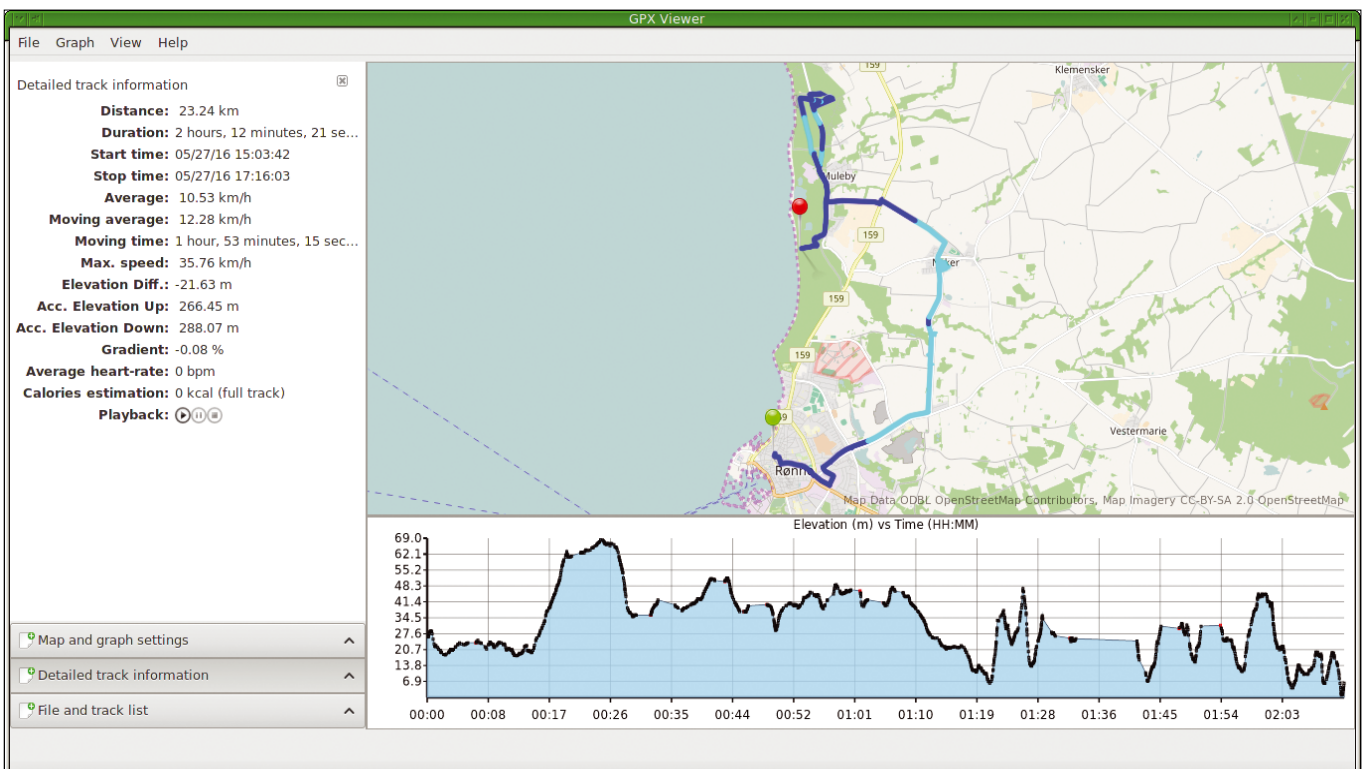


Figure 6: The GPX Viewer follows a track virtually, showing the exact position, speed, and breaks.

We test drive the latest Skype for Linux

Speak to Me!

The old Microsoft gave the cold shoulder to Skype for Linux, neither updating it nor equipping it with new features. The new Microsoft promises to do better, with a brand new alpha prerelease version of a Linux client for the famous telephony tool. *By Heike Jurzik*

Skype [1] has been around since 2003, and Microsoft took over the service in 2011. The free program supports instant messaging, IP telephony, video conferencing, data transmission, and screen sharing. Skype versions for Android, iOS, and Windows Phone are also available. Even though Skype uses a proprietary protocol, and it is not considered secure, it has remained the leading IM and telephony solution for many users, partly because it is available for all major operating systems and platforms.

Linux users, however, have been out in the cold since 2014. The last version of Skype that runs on Linux (Skype 4.3 for Linux) only supported 32-bit architectures, and many Linux users experienced problems getting it to work consistently. Skype 4.3 for Linux installed on many Linux distributions with a modicum of effort, but it was missing important features such as video conferencing. On July 13 of this year, Microsoft announced a new alpha version for the Linux desktop in a blog post [2]. Microsoft's newfound affection for Linux was evident in the announcement, which stated, "This release reaffirms our commitment to an important group of people – the Linux community."

Version 1.2.0.1 of Skype for Linux is still in an early stage of development. Many features haven't been implemented yet, but you can already experience some important improvements, such as a faster and more feature-rich Skype UI. Microsoft says it released the alpha version to get feedback from the community, so we decided to start it up and try it. We tested the new Linux client on Ubuntu 16.04 (Unity) and openSUSE Leap 42.1 (KDE Plasma), and we also phone-called Windows, OS X, and iPhone users.

Microsoft says it will release a new version of the alpha prerelease every two weeks, so by the time you read this article, new versions might already be available that address some of the issues highlighted in this article.

Ring, Ring

Debian and RPM packages are available in the download section, each for 64-bit systems [3]. During installation, both packages create a new repository named `skype-stable` [4] and resolve various dependencies, including adding the Gnome Keyring (`gnome-keyring`) package if it is not already installed. Skype does not work with KWallet or other password managers.

After you launch `skypeforlinux`, enter your access details, or create a new user account by selecting **Create new account**. Although the software is nicely localized for other platforms, as things stand now, the alpha version of the Linux edition only speaks English.

As usual, clicking on your avatar top left takes you to your account settings, where you can change your profile picture, status, or add a personal message. Windows and OS X clients provide many more options, allowing you to enter phone numbers, location details, email addresses, gender, and date of birth. In the alpha version of the Linux client, you can only configure audio and video devices during an ongoing conversation via the gear icon in the main window.

History, contacts, the Skype configuration, and a button for new conversations are accessible in the left sidebar. You can search for other Skype users and contact them using **Search Skype**. Our test of the search feature yielded some varied results; some requests succeeded in finding users, and others did not. For testing purposes, our team confirmed contact on OS X and hoped that the Linux client would accept the information. Synchronization succeeded, but only after restarting Skype.

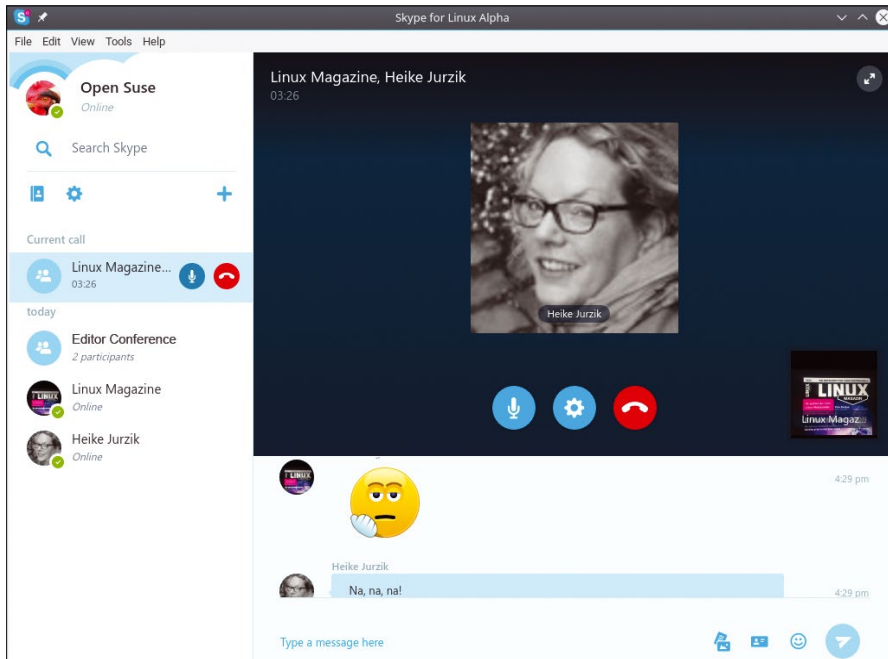


Figure 1: The test team in the audio conference: The avatar of the user currently speaking appears highlighted in the main window.

Instant messaging already runs smoothly between two users or a group of users, and the (sometimes rather annoying) Skype emoticons are available for Linux users. In one-to-one chats, clicking the globe calls Skype Translator, which can translate into over 50 languages. Functions that scored well in the test include a feature that shows whether the other person is typing (if the other person has allowed that information), along with data transfer, contact

sharing, and invitations to chat. The Linux client failed to play video messages, showing just a thumbnail.

Service Hotline

The alpha version of Skype for Linux dropped a few simple phone calls. Although discussions from Linux with iOS and OS X users worked, the microphones often remained mute for calls between Linux and Windows clients. The testers repeatedly saw the missed call

(Linux) and sorry, we couldn't connect (Windows) messages. Occasionally, it was actually possible to leave a voice message.

An audio conference between two Linux clients and the iOS user proceeded without any problems (Figure 1). In the alpha version, it is no longer possible to add contacts to an ongoing conversation session; users must first create the group and then begin the call.

Some of the planned features have not yet been implemented in the alpha version. In addition to video calls and conferences, these features include screen sharing, calls and text messages to a landline or cellular network, and an option for managing your Skype credit balance. Those functions are likely to make their entrance quite soon, with a new release appearing every two weeks [5]. The updates should be available in package form; see the announcements and discussions at the Skype Community site for more information [6].

If you like, you can leave feedback for the Skype programmers and include your machine's logfiles by enabling debugging mode with the [CTRL] + [Shift] + [ALT] + [D] key combination (Figure 2). If you are interested in getting in touch with the creators, leave a message through the *Help | Feedback* menu item, or use the *#skypefeedback* hashtag on Twitter. ■■■

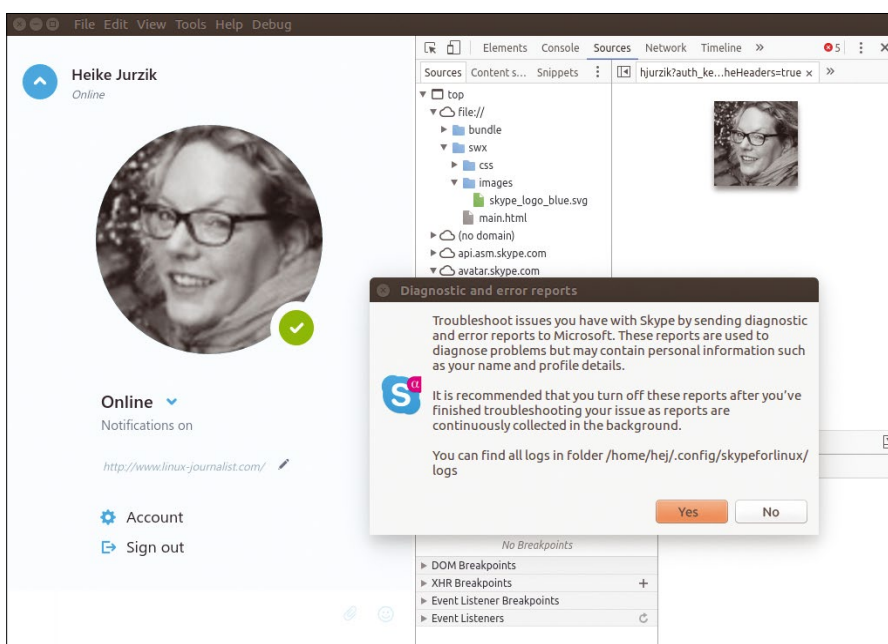


Figure 2: You can turn on debugging mode and logging for all Skype messages from the Debug menu.

INFO

- [1] Skype: <https://www.skype.com>
- [2] Blog article on Skype for Linux: <http://blogs.skype.com/2016/07/13/skype-for-linux-alpha-and-calling-on-chrome-and-chromebooks>
- [3] Skype alpha version download: <https://community.skype.com/t5/Linux/Skype-for-Linux-Alpha-and-calling-on-Chrome-and-Chromebooks/td-p/4434299>
- [4] Skype sources: <https://repo.skype.com>
- [5] Skype alpha version support: <https://support.skype.com/en/faq/FA34656/more-information-about-skype-for-linux-alpha>
- [6] Skype Community: <https://community.skype.com/t5/Linux/bd-p/Linux>



Putting office suites to the test

Light and Shadow

In the office, the interoperability and cooperation of a few programs play an important role. We take the four big Linux office suites to task and see how well they cope with non-native formats. *By Erik Bärwaldt*

Office suites are probably the most frequently and universally used programs; therefore, every desktop Linux distro includes them when installing on a hard disk. Although the features of the free word processors, presentation programs, and spreadsheets outside the Microsoft world differ hardly at all, the integrated programs aim to score points with additional features and unique operating concepts.

Two key questions that determine the success or failure of an office product are: How compatible it is with Microsoft formats? Do providers within the industry support it?

LibreOffice 5.0.6 [1], Apache OpenOffice 4.1.2 [2], SoftMaker Office Professional 2016 [3], and WPS Office 10.1.0.5672 [4] demonstrate their practical capabilities in different environments that developed over time (Table 1). Along with interoperability, I look at the user interface, because the best conversion filters for foreign file formats are of little use if the office program is difficult to control.

Commenting in text documents is the order of the day at many companies and needs to work intuitively and between packages. Here, I look at how well the office suites meet current standards and how quickly the average user can come to grips with any divergence or import artifacts.

Last, but not least, the test also considers how well the test candidates fit into existing business infrastructures.

The usefulness of an office application hinges on the interfaces provided for it by third parties so that it can be part of an integrated solution.

Format Chaos

On Windows and OS X, the operating systems leading by market share, Microsoft Office has firmly established itself as the standard. The Redmond-based giant has not only introduced many new versions of Office over the years but has also modified the file formats. Office documents are not all in the same boat. With no backward compatibility, users cannot open new MS Office documents on older versions.

The alternative office suites must support a variety of MS Office formats to be useful universally, which means that developing new conversion filters is far from a trivial programming task. Three different, mutually incompatible standards exist for Office Open XML (OOXML), and the format specifications span more than 6,000 pages [5].

Another problem stems from the various scripting and macro languages for programming embedded in documents: Whereas Microsoft relies on its VBA (Visual Basic for Applications) dialect, LibreOffice and OpenOffice both use Basic code, albeit with a different methodology and separate objects. The other contenders provide their own Basic dialects as macro and scripting languages, mean-

ing that documents featuring macros in MS format usually will not run smoothly without intervention.

Apache OpenOffice

First into the ring is Apache OpenOffice [2]. Originating in 2000, this office suite is one of the oldest alternatives. The first versions were completely based on the source code by StarOffice, which Hamburg-based Star Division had developed in the 1980s.

During its varied history, OpenOffice passed to Sun Microsystems with the sale of the company in 1999, then to Oracle in 2009. In recent years, this former shooting star among office suites has massively lost out on market presence, and the developer community has shrunk significantly after Oracle delivered a huge snub to many of the very dedicated contributing programmers with restrictive licensing terms in 2010. They migrated in droves to the newly founded Document Foundation and worked to improve the LibreOffice office suite derived from OpenOffice.

Although Oracle transferred the rights for OpenOffice to the Apache Software Foundation [6] in 2012, it was not possible to make up for the drain of qualified staff, meaning new releases of OpenOffice have not been forthcoming. The current version 4.1.2 has been out since October 2015 [7]. Apache OpenOffice is available for Linux as a precompiled tar.gz archive for 32- and 64-bit architectures [8]. The archive, only around 170MB in size, contains RPM or DEB packages, depending on the platform (32- and 64-bit), and is offered in many languages.

In the test, however, the 64-bit variant flunked on the ROSA Desktop Fresh R7

TABLE 1: Comparing Office Solutions

Features	OpenOffice	LibreOffice	SoftMaker Office	WPS Office
License	LGPL	Mozilla Public License	Proprietary	Proprietary
Version	4.1.2	5.0.6	Professional 2016	10.1.0.5672
Price	Free	Free	\$70, EUR60 (Standard)	Free (Linux), \$80, EUR77 (Professional)
Word processing	Yes	Yes	Yes	Yes
Spreadsheet	Yes	Yes	Yes	Yes
Presentation	Yes	Yes	Yes	Yes
Database	Yes	Yes	No	No
Drawing program	Yes	Yes	No	No
Formula editor	Yes	Yes	No	No
Spell check	Yes	Yes	Yes	Yes
Additional dictionaries	No	No	Yes	No
Own macro language	Yes	Yes	Yes	Yes
Template management	Yes	Yes	Yes	Yes
Export				
PDF	Yes	Yes	Yes	Yes
EPUB	No	No	Yes	No
MS Office	Yes	Yes	Yes	Yes
ODF	Yes	Yes	Yes	No
Import				
MS Office	Yes	Yes	Yes	Yes
ODF	Yes	Yes	Yes	No

Linux distro: The package installed, but I was unable to launch it. Because I failed to find any meaningful log entries for the cause of the glitch, I left Apache OpenOffice out of the test on the Russian Mandriva edition.

After a successful installation, you will find some new entries in its menu structure in the All Programs subfolder: OpenOffice creates its own starter for each individual module, and another for the selection screen, from which interface you select the individual applications (Figure 1).

As a full-fledged office suite, in addition to the three standard applications – word processing, spreadsheet, and presentation – OpenOffice also includes a database and a built-in formula editor for mathematical and scientific formulas. A vector-based drawing program is also integrated in the form of Draw. The individual components of the office suite can link and exchange data with one another; external conversions are not necessary.

Interface

Apache OpenOffice offers a contemporary interface without frills and gim-

micks. Since IBM handed over the source code of its office suite Lotus Symphony to the Apache Software Foundation two years after discontinuing the product in 2012, the free office program has benefited from Symphony’s innovative operating concept. As one of the first major office programs, OpenOffice was equipped

with a collapsible sidebar on the right side that took advantage of modern wide-screen displays much more effectively than older office applications [9]. The context-sensitive sidebar (Figure 2) provides formatting, templates, and graphics and removes the need to access the traditional menu bar. If desired, you can integrate frequently used items into the symbol bar, which is arranged horizontally at the top of the program window, and you can remove rarely or never needed features. All the other candidates offer this feature, as well.

File Formats

Apache OpenOffice uses version 1.0 of the Open Document Format (ODF) standardized in 2006 as its default file format. The office suite supplies a decent number of conversion filters, as well: The Writer word processor can import and export all current Microsoft Word formats, in addition to simple text formats. However, in the practical test, OpenOffice fell short in this respect. Although the word processor did import sample documents in the older Microsoft Word format (.doc extension), the outcome remained unsatisfactory, and the entire layout of the integrated images and graphics of each document was changed.

The font processor in OpenOffice is a further weak point. If certain fonts are not installed, the office suite replaces them with similar fonts, which can cause translocation of paragraphs and incorrect line breaks because of spacing differences.



Figure 1: OpenOffice provides a selection screen on startup.

You are best advised to install the relevant fonts globally before loading the documents – assuming the fonts are known.

The free office suite also trips over tables when importing foreign formats. After processing a table, changed font sizes sometimes overwrite or displace columns and rows, making the entire table unreadable. You can resolve this problem by highlighting the sections and reducing the font size.

OpenOffice also encountered difficulty in the complex document test with the OfficeOpen XML format by Microsoft introduced in 2007 (.docx extension). When importing a spreadsheet for a school class, it completely misplaced images, graphics, and grids, making the document virtually unusable (Figure 3).

Users of the old StarOffice, who still use documents in .sdc, .sdd, and .sdw formats, can no longer read and write these files as of Apache OpenOffice 4.0. In this case, the solution is to install a “portable” older version of the office suite [10] that can still import and export these formats. You can install the portable version along with the current version on your local disk; however, you should note that the tar.gz archives of the portable variants are not available for all hardware architectures on Linux.

Collaboration

Apache OpenOffice does not have trouble reading documents in non-native formats when the comments of several editors

are integrated. It presents the comments very clearly on the right of the screen with a reference, the name of the respective author, and different color schemes. You can process each comment from a small selection menu.

Interfaces

Because of its cross-platform availability and widespread popularity, many special industry solutions support OpenOffice in custom interfaces. The free office suite easily integrates into the major league enterprise resource planning (ERP) solution by SAP through an interface with its in-house MaxDB database back end [11].

Actiware, which manages, analyzes, and exports Actiwatch data, offers an integration solution in the form of Awas-Actiware Active Server [12]. Its Office integration module cooperates with virtually every enterprise solution, irrespective of provider, guaranteeing the transfer of data between business software and the office program.

LibreOffice

The Document Foundation maintains LibreOffice [1], the product that was forked from OpenOffice in 2010, and is developing it further. The office suite is currently the default in almost all major Linux distributions and is available across platforms. Although the original codebase of OpenOffice and LibreOffice is almost identical, the two office programs have diverged over the years because of licensing rights issues.

Both suites, however, use the Open Document format by default, and it is even possible to work simultaneously with both office programs in different environments without a problem. However, LibreOffice is now showing far more agility than OpenOffice, and the community also provides better support.

Interface

LibreOffice 5.0.6 offers an interface that is functionally similar to that of OpenOffice, which differs only in appearance. It adopts the OpenOffice sidebar in a modified form.

Unlike OpenOffice, in which users always install all the components, Linux distributions sometimes only load individual components – primarily the Writer word processor, Calc spreadsheet, and Impress presentation program. The software repositories of many Linux distros

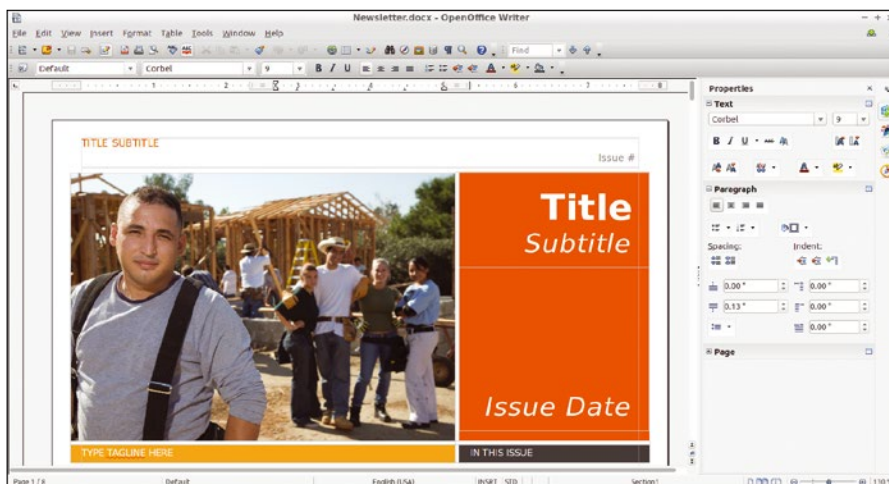


Figure 2: The Apache OpenOffice 4.1.2 interface uses available monitor space with optimal efficiency thanks to the sidebar.

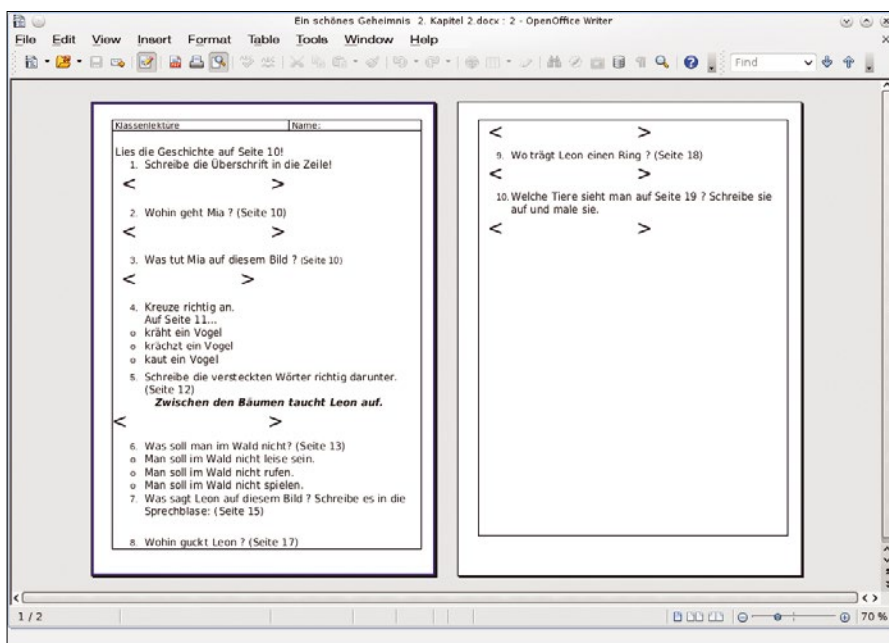


Figure 3: Failed: Complex DOCX documents overwhelm OpenOffice.

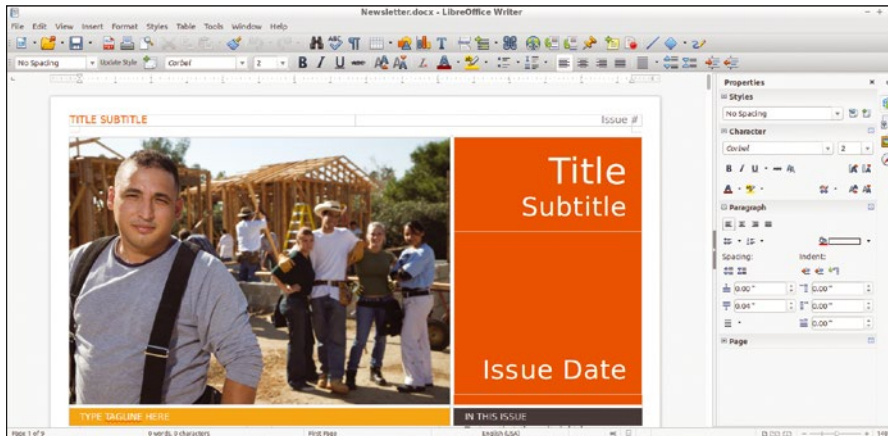


Figure 4: LibreOffice and OpenOffice operate very similarly.

then have extensions that enhance the office program's feature scope, which you have to install manually in OpenOffice.

Because of the great similarity between LibreOffice and OpenOffice, those switching to LibreOffice will hardly need any training to make the move (Figure 4) – even the suite's program modules have the same names as those in OpenOffice. In direct comparison, however, it is striking that LibreOffice lets you work much more smoothly. Not only do the individual modules launch faster, large files also open at a noticeably quicker pace, as well.

A Question of Formatting

When tested with intricately designed Microsoft Word files, LibreOffice was not fully convincing in its handling of either the older DOC or newer DOCX variants. Much like OpenOffice, it sometimes made errors in displaying fonts, locating placeholders in the document, and, in a few places, rendering the table layout. Even so, LibreOffice's conversion filters were able to display graphics in the document (Figure 5). Although it flawlessly imports simply organized documents, you still have to put in some time-consuming manual rework for layouts with a more complex structure.

Compared with the Apache product, LibreOffice gains points with its considerably larger number of import and export filters that can load some of the more time-worn formats without error. Additionally, it has an integrated export filter for the widespread PDF format, which is well-crafted and indispensable for office use.

Comments

When it comes to the comment function, which is important for collabora-

tive work on documents, LibreOffice falls short compared with Apache OpenOffice. It highlights comments in color to reflect the annotator and refers correctly to the relevant passages in the original text. However, the office suite trips over existing comments in some places and fails to display them. This problem especially occurs at page breaks. Moreover, LibreOffice occasionally changes the typeface within the comments for no reason, which is likely to confuse readers. These bugs significantly reduce the benefits of the office program for team activities.

Interfaces

The standardized Open Document Format used by LibreOffice allows it to use the interfaces originally written by third-party developers for OpenOffice. In most cases, then, the office suite can connect without a problem to corporate ERP solutions originally designed for OpenOffice.

SoftMaker Office

The third candidate to face the test is SoftMaker Office Professional 2016 [3] from the Nuremberg-based software house of the same name. The proprietary office program is available in Standard and Professional editions for multiple platforms and has now reached version 757.0510 [13]. In addition to the office program, the company offers various font packages.

SoftMaker Office can be traced back to 1987, so it is one of the dinosaurs in the world of office programs. The suite includes applications for the three most important areas of use: the TextMaker word processor, PlanMaker spreadsheet, and Presentations presentation program. The DataMaker database offered in previous years is no longer on board. Linux has had a version of the package since 2003, since which it has undergone several product cycles.

The office suite is available as a 30-day trial version in a binary package approximately 325MB in size for systems with DEB or RPM package management. Another variant has a shell installer. You can install the trial version permanently after purchase by entering a license number.

Interface

The SoftMaker Office interface design looks much like the office suites already discussed, even as far as sporting a sidebar to make better use of widescreen monitors. Compared with the previous two free office solutions, the speedy pace of SoftMaker Office is striking. The individual modules are not only quicker to launch, the suite also scrolls through

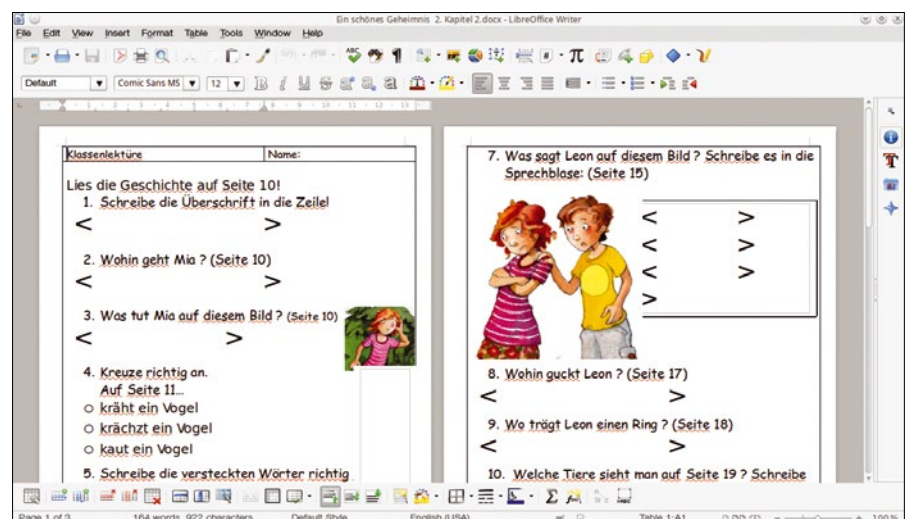


Figure 5: LibreOffice gets along better than OpenOffice with DOCX documents – but, again, not really well.

long documents more capably and completes its searches in document or large tables in no time.

Language Support

SoftMaker Office has all the functions necessary for daily use, and you can customize individual features to tweak the appearance. The office suite only reveals some of its unique selling points when you look through the menus: In the Standard edition, SoftMaker Office provides commercial dictionaries for 20 languages and thesauri in eight languages, and the Professional version adds Berlitz dictionaries for translations to and from English, German, French, Italian, and Spanish (Figure 6).

Specifically, without installing additional tools, users of different nationalities have dictionaries of foreign loan words and universal dictionaries at their disposal in English, French, Spanish, German, and Italian for translation support. You do not need any training to use the intuitive interface. For spelling corrections, SoftMaker Office uses the same free Hunspell checks found in OpenOffice and LibreOffice.

The office suite from Nuremberg uses an in-house file format by default, although it also imports and exports the most popular formats from the Microsoft realm. SoftMaker Office generates simple text files as well as ODF documents, so it can also exchange data with LibreOffice and OpenOffice. The program additionally exports existing documents into PDF and the open EPUB formats (for e-book readers).

SoftMaker Office fared especially well in the lab when processing more complex

and older MS Office documents. Manual intervention was not required for demanding layouts or extensive tables, and it replaced unavailable fonts with more suitable alternatives than its competitors.

Yet it stumbled when it came to reading documents properly in OOXML format. Graphics, tables, and special paragraph formatting were broadly imported correctly, but placeholders were not fully applied. Predefined symbols also posed problems for the conversion filter. However, manual rework was kept within limits, meaning that this office suite pulled ahead of LibreOffice and OpenOffice (Figure 7).

Teamwork

The SoftMaker Office conversion filters also accept comments in documents from other file formats. However, the way these are displayed takes some getting used to: As with other test subjects, it displays the comments to the right of the main text with a color-highlighted reference, although they disappear behind the sidebar. To display comments without having to scroll horizontally, you can collapse the sidebar with *View | Sidebar | Hide*.

As with the other office suites, you can edit existing comments in a context menu that appears after right-clicking on the comment. SoftMaker Office has fewer context-sensitive actions from which to choose, although it offers a number of formatting options that are of little help in annotations.

Interfaces

Any application that offers an interface for Microsoft Office, OpenOffice, and

LibreOffice can also accommodate SoftMaker Office documents. If the templates provided by a third-party provider include macros or scripts, SoftMaker Office typically does not have an answer – even if the templates are in a supported file format – because of incompatibilities between the various dialects of Basic.

WPS Office

Largely unknown in the Western world, WPS Office [4] comes from China and was created by Kingsoft. The Linux version of this office program, which was developed for cross-platform sharing, has officially enjoyed alpha status for several years, and the developers constantly maintain and develop it.

Several releases back, the developers started to focus on bugfixes and are no longer implementing new features, meaning that the suite is now extremely stable and works reliably. Alpha version 21, which is currently available, focuses primarily on completed localizations of the software package. Like SoftMaker Office, WPS Office includes a word processor called Writer, and the self-explanatory Spreadsheets and Presentation modules.

The Chinese office program is available for Linux in the current Intel 32- and 64-bit architectures in the form of DEB and RPM files for package management systems [14]. Additionally, the creators offer a portable tar.xz archive that can run on almost any Linux system. Although the pre-compiled DEB and RPM packages take up roughly 78MB, the portable version weighs in at 150MB.

Appearance

Compared with the other three subjects in the test, WPS Office has a radically different appearance, with a different operating concept. It resembles a modern web browser, in that the software opens a separate tab for different documents.

On the home screen, WPS Office displays a graphical template view, from which you can choose pre-made templates. This view, known as Docer Online Templates, downloads template views in a context-sensitive way from the Internet, depending on the program module used (Figure 8). If you want to use a template or view it in more detail, it is displayed in an overlay window when clicked. If you so want, you can have WPS download

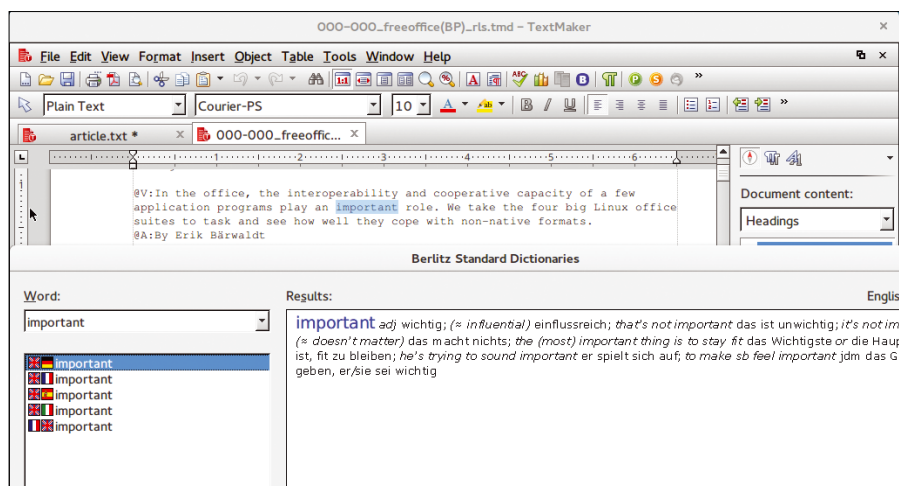


Figure 6: Integrated reference works help SoftMaker Office stand out from the crowd.

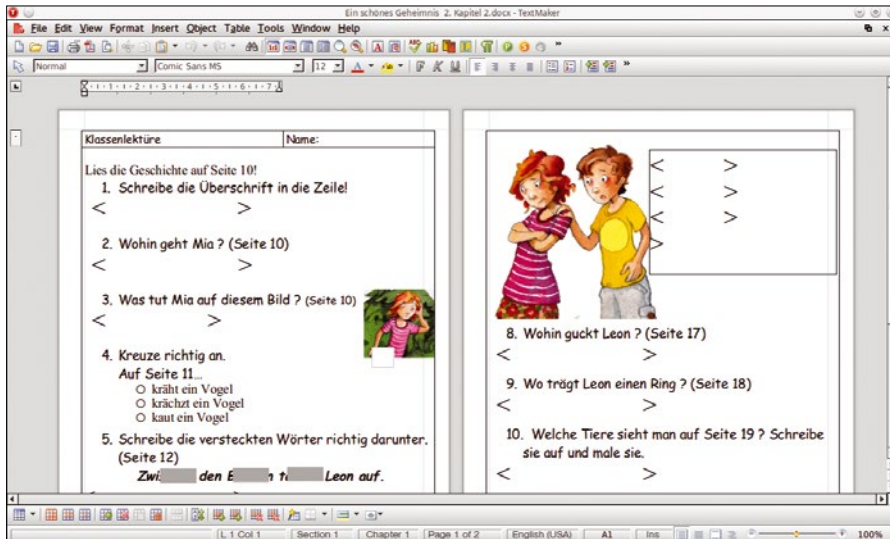


Figure 7: For complex DOCX documents, SoftMaker Office also demonstrates a need for improvement.

the formatting templates by clicking on the *Download* button at the bottom right of the preview window.

If you want to open an empty window without formatting templates, click on the *New blank document* button in the top right of the program window. The blank document then appears in a new tab.

Chameleon

The Chinese office suite also reveals itself to be extremely visually versatile. The *Tools | Switch UI* menu allows you to choose between several tile views of the interface or the classic layout; you must restart the respective program module to activate the selected interface. You can use the individual modules with different interfaces, because the selection only applies to the active module. This minimizes the learning curve for newcomers who stick to using the interface they are familiar with from the office package they previously used.

WPS Office is also very flexible and innovative when it comes to language support: The community is currently completing localization tasks, but the package is a

delight, with its tremendous support of foreign languages. You can change the language with just a few clicks of the mouse from the *Tools | Switch Language* menu.

When you select the desired language from a new dialog, the system downloads the relevant language files from the web. WPS enables the new localization after restarting the respective program module (e.g., Writer, Presentation, etc.).

Unlike the visual themes, the language selection affects all modules of the suite, meaning you do not need to adjust the language in each and every module. WPS Office simultaneously stores several localized versions, so you can switch between languages in the language dialog box with a mouse click and module relaunch (Figure 9).

Similarly unusual is the way WPS Office manages backup copies. If the program terminates unexpectedly, it can still access the active document. After restarting, *Tools | Backup Management* will bring up a list view in the sidebar that displays all of the backup files. You can select the desired recovery file there.

File Formats

The Chinese office suite claims to be fully compatible with Microsoft Office documents. Although WPS Office has its own file and template format, it naturally provides conversion filters for the import and export of content in various Microsoft formats.

The WPS suite also offers filters for templates with built-in macros, the only contestant in this test to do so. Microsoft saves macros with a *.dotm* or *.docm* extension in the new versions of its Office program. Nonetheless, in many cases, WPS Office has to throw in the towel with Microsoft documents containing macros or scripts.

In my test with complex templates, WPS Office did not achieve full compatibility with its conversion functions for MS Office formats, although it had by far the best result compared with the other office suites, especially when taking on OOXML documents. Graphics, tables, and placeholders appeared in the appropriate places, with fonts rarely switched.

Beyond the Microsoft world, the WPS Office modules offer hardly any options for exchanging data, however. The suite only offers filters capable of handling web pages and the simplest text files without formatting. The office suite is

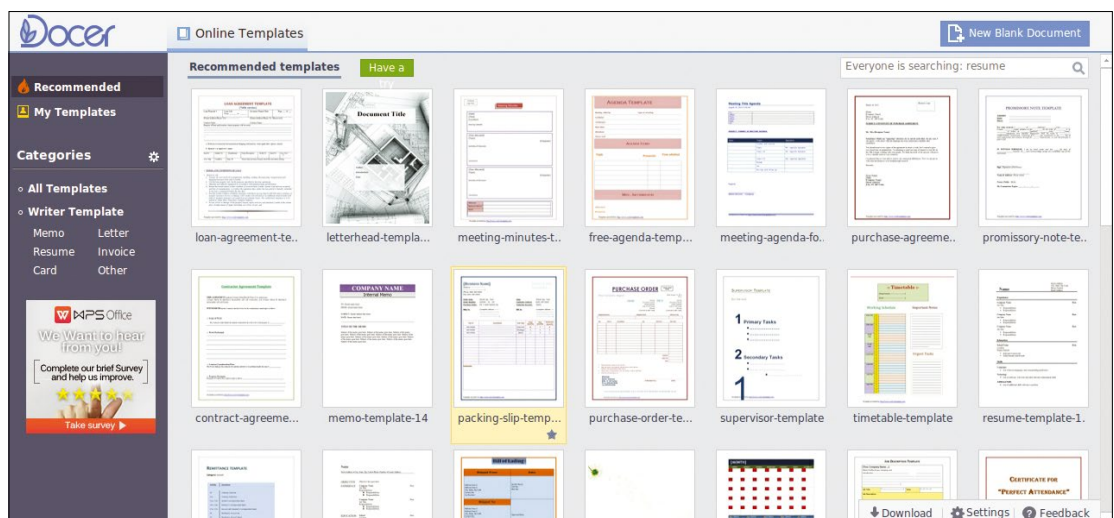


Figure 8: Unusual, but quite understandable: The WPS Office interface.



Figure 9: A few clicks change the language in WPS Office.

apparently unfamiliar with the Open Document Format, although it still recognizes the dBASE database format, which in fact is no longer in use. The DIF format, also very dated and almost irrelevant in modern IT infrastructures, is still supported by the spreadsheet application (Figure 10).

Teamwork

WPS Office also performs pretty well with documents to be edited by multiple users. The office suite from Beijing is the only product in the test that highlights the comments and their related references and labels completely and with the correct colors. Simple symbols in the status lines of individual comments also make it easier to edit or delete annotations. You can create new references and comments with the *Insert | Comment* menu item,

with many enterprise applications, however, via the import and export filters for Microsoft files. It would surely be even better for users and useful for companies if the source code were available.

Conclusion

This review shows that major office suites available on Linux are functionally the same as Microsoft's top dog, although interoperability in terms of non-native file format support leaves plenty to be desired.

The increasingly used OOXML format by Microsoft presented serious problems for all of the test candidates, with the exception of WPS Office, particularly with complex documents. The import and export filters of the two large free office suites, LibreOffice and OpenOffice, proved inadequate in the test – so much

and the software automatically selects the relevant colors for new commenters, making it clear which person is the source of each comment.

Third-Party Support

So far, not unexpectedly, no industrial or commercial applications from Europe support the native file formats of the exotic species from China. WPS Office is able to communicate

so that it seemed less time consuming to completely rebuild the document from scratch than to correct the erroneous formatting.

In terms of collaboration, except for OpenOffice and WPS Office, all candidates still have a lot of catching up to do if the documents originate in the Microsoft universe.

In the practical test, the SoftMaker Office Professional version was able to score points because it integrates dictionaries and can export documents to EPUB format. WPS Office, however, has the best conversion filters, but users will need a little more training because of some innovative concepts.

On the other hand, the free office suites impress, because industry software support is far superior to that for the two solutions by SoftMaker and Kingsoft. They also can connect to commercial ERP software without a problem. ■■■

INFO

- [1] LibreOffice: <https://libreoffice.org>
- [2] Apache OpenOffice: <https://www.openoffice.org/>
- [3] SoftMaker Office: <http://www.softmaker.com/en/softmaker-office>
- [4] WPS Office: <https://www.wps.com/>
- [5] Office Open XML Specification (OOXML): <http://blogs.fsfe.org/greve/?p=123>
- [6] Apache Software Foundation: <https://www.apache.org>
- [7] Current AOO release: <https://cwiki.apache.org/confluence/display/OOoUSERS/AOO+4.1.2+Release+Notes>
- [8] AOO download: <http://www.openoffice.org/download/index.html>
- [9] Symphony in OpenOffice: <http://www-01.ibm.com/support/docview.wss?uid=swg21661339>
- [10] AOO portable versions: <http://www.openoffice.org/download/index.html>
- [11] SAP integration: https://help.sap.com/saphelp_sem60/helpdata/en/fd/14f34026d70b06e1000000a155106/content.htm
- [12] Activware modules: <http://activware-gmbh.software.informer.com>
- [13] SoftMaker versions: <http://shop.softmaker.com/shop/shop.php>
- [14] WPS Office download: <http://wps-community.org/download.html>

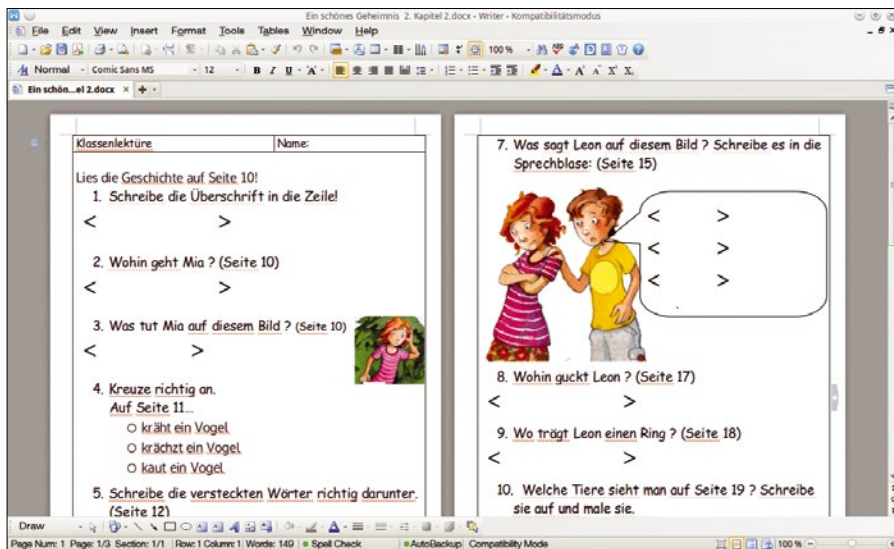
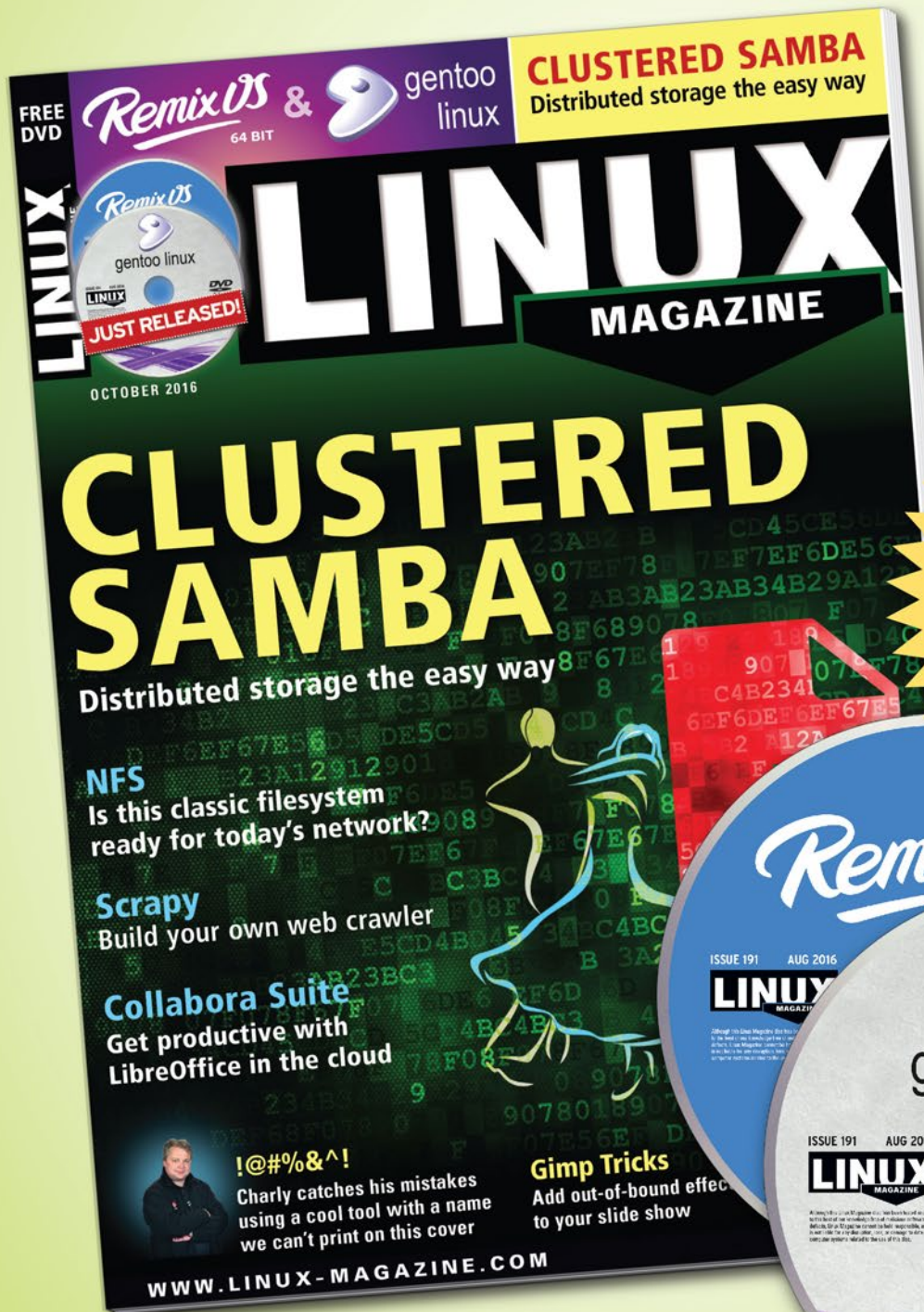


Figure 10: Almost perfect: Handling Microsoft documents without macros and scripts in WPS Office.

Subscribe now!



Don't miss a single issue of the magazine that delivers the in-depth technical solutions you'll use everyday!



shop.linuxnewmedia.com/subs



Gimp and neural networks

Matching Paint Job

Deep learning isn't just for industrial automation tasks. With a little help from Gimp and some special neural network tools, you can add color to your old black and white images.

By Sebastian Mogilowski, Markus Feilner, and Kristian Kißling

Neural networks (NN) don't just play the traditional Japanese board game Go better than the best human player; they can also solve more practical tasks. For example, a project from Japan colors old black-and-white photos with the help of a neural network – and without asking the user to get involved with the image editing.

Researchers at Waseda University in Tokyo used a database that contains several objects to train a neural model to correctly recognize objects in images and fill them with appropriate color information. Using this model, the network then identifies the individual parts of the image, say, trees and people, and assigns matching colors.

The Waseda team presented this deep learning tool at the SIGGRAPH 2016 computer graphics conference [1]; you will find the code for their photo-coloring tool on Github [2]. The university website [3] provides a research paper on the subject [4], as well as some sample images.

Neural networks consist of many layers that gradually filter out information. For an image, that image might consist of the brightness, edges, and shadows. At the end, the network identifies specific, complex objects. Siri, Google Now, or Cortana use the same principle for speech recognition.

The problem with a conventional neural network is that each layer can make mistakes. The layer then passes

this mistake on to the next layer. The type of neural network used with the tool described in this article, which is called a convolutional neural network (CNN) [5], has some built-in ways for limiting the effects of errors.

CNN versus NN

The concept for CNNs comes from biology, although it is not the human brain that serves as a template, but the visual cortex of cats. The Convolutional layer considers the spatial structure of the objects.

CNNs differ from conventional neural networks in the type of signal flow between neurons. The signals in NNs typically pass along the input-output channel in one direction, without the

Lead Image © Orlando Rosu, 123RF.com

ability to iterate through a loop. CNN takes a different approach. The areas on which the neurons operate overlap and are arranged with offsets. Each layer contributes to a more reliable overall result, thus optimizing the detection rate. The network can identify an object even if it is different from the position defined by the training templates.

Deep learning makes it possible for a computer to identify the objects in an image. This procedure works even when the object on a screen is significantly changed compared to the training model, say, because it has a different background or because the viewing angle of the object or the lighting conditions have changed [6].

Tasks that require visual recognition are something that CNNs cope with very well. But the result depends on the quality and amount of training data, as you will see in the sample pictures later in this article.

The model shown in Figure 1 consists of roughly four parallelized and combined networks. The low-level features network recognizes the corners and edges of an image in high resolution. The data ends up in the global features network, which then sends them through four convoluted layers and then through three fully connected layers that each link the neurons of the one layer with all of the neurons in the other layer.

The result is a global, 256-dimensional vector representation of the

image. On the other hand, the mid-level features network extracts textures from the data from the low-level features network.

The results of the global and mid-level-features network are then combined in the fusion layer; the results are resolution-independent thanks to vectorization. Finally, the colorization network adds color information (chrominance) and luminance and restores the resolution of the source image.

The end to-end network thus brings together global and local properties of images and processes them at any resolution. If the global features network suggests that the shot was taken outdoors, the local features network then tends to focus on natural colors.

Not Only Gray Theory

You can use the software from the Japanese researchers and the GIMP image-processing tool to colorize black-and-white images. You'll need a powerful computer with a reasonably recent graphics card.

In my test, I used Ubuntu 16.04 with Gnome. (The Japanese team used Ubuntu 14.04 with Gnome.) To follow the examples, you need to install Git, Gimp, and the LUA package manager, Luarocks:

```
sudo apt-get install git gimp luarocks
```

With only marginally more effort, you can then install Torch [7], Facebook's deep learning library. Torch is written in

Lua [8] and is available under a BSD-style license. The Torch library provides algorithms for deep learning and is easy to install thanks to LUA.

Because Torch uses C backends and a MATLAB-like environment, it is perfect for scientific projects. Torch also includes packages for optimizing graphical models and image processing. The associated nn package produces neural networks and equips them with various abilities.

You can clone Torch yourself from Github. Finally, you need to execute the included install script:

```
git clone https://github.com/torch/distro.git
~/torch --recursivecd
~/torch
bash install-deps
./install.sh
```

The step adds Torch as a PATH variable to your .bashrc; you will want to restart Bash at this point. Now, you need to install some LUA packages on the computer:

```
luarocks install nn
luarocks install image
luarocks install nnglyph
```

The next step is to set up the actual coloring software. You can download this software from Github [2] using git clone; I used the supplied script named download_model.sh to install on my machine:

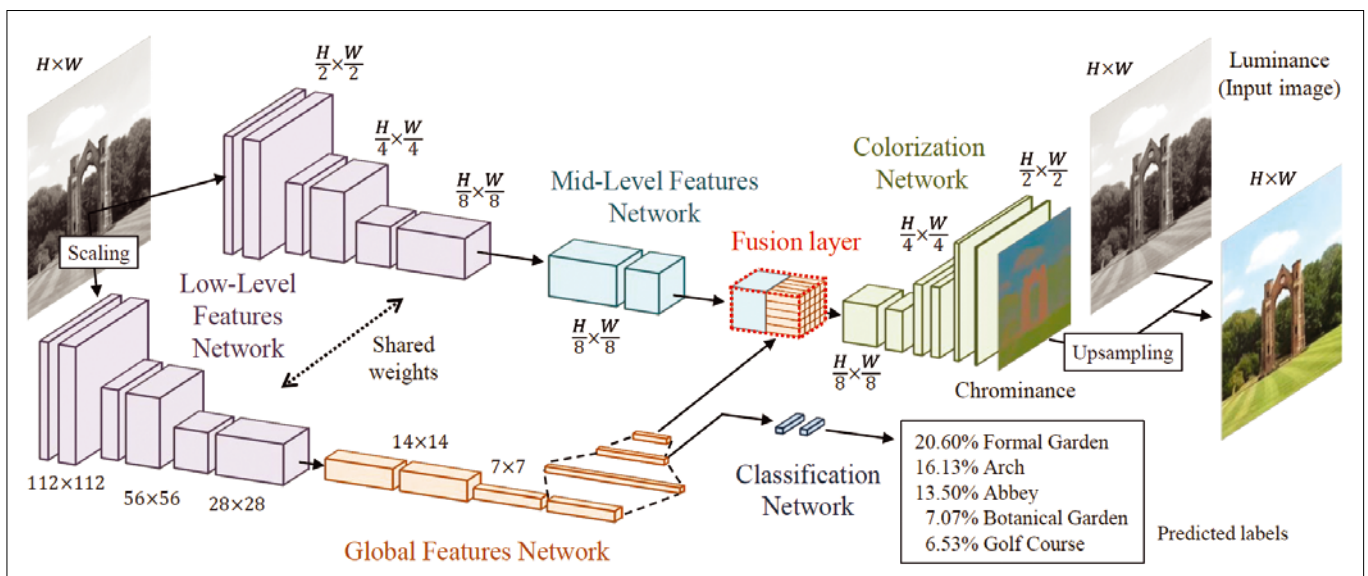


Figure 1: Put simply, the model used is a network that can be divided into four functional networks.

```
cd ~
git clone https://github.com/satoshiizuka/siggraph2016_colorization.git
cd siggraph2016_colorization
./download_model.sh
```

For my first attempt, I copied the test1.jpg image to the siggraph2016_colorization folder. The test image is a scan of a photo with 638x638 pixels. I trimmed

the image to a square shape because the neural network was trained on square images. Then I handed it over to the colorization script:

```
th colorize.lua test1.jpg test1_color1.jpg
```

The not-entirely-so-convincing first results are shown in Figure 2. This uninspiring result is probably due to the fact that the

For my next attempt, I used GIMP to convert the image to grayscale (Image | Mode | Grayscale); this change did not cause any visible changes to the image. I then reduced the image to 224x224 pixels but without grayscaling. This step affected the resolution, but at least it improved the color scheme. Finally, I set up the grayscales and reduced the image to 224x224 pixels (Figure 3). But how do I transfer the significantly better color information to pictures with a higher resolution?

New Layer

GIMP lets you break down an image to create a Lab color model [9]. GIMP divides the image into three layers:

An L level for the luminance, an a level for the hues between green and red, and a b level for the colors between blue and yellow [10].

The idea is to break down the large and small images into this color space. Then I would scale the a and b layers of

CNN processes images 224x224 in size. Also, my image template does not really consist of grayscales.



Figure 2: A first colorization run is not very effective; the image is still not optimized for the colorization software.

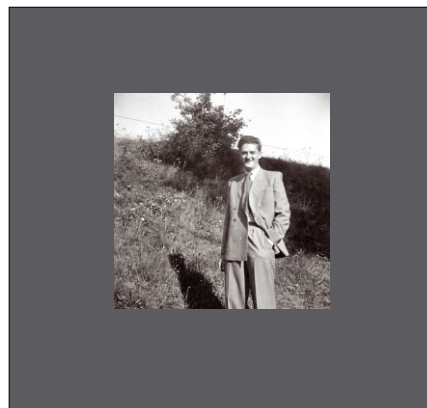


Figure 3: Grayscale and scaled to 224x224 pixels: Better colorization – bad resolution. GIMP to the rescue.



Figure 4: The small image in the original, and next to it the lab color layers extracted via GIMP (top left).

the small image scale to the resolution of the larger image and transfer them to it. When you put the levels together again, you get the larger picture with the color information from the smaller image.

To do this, you open the small colored image and the large picture in GIMP as the first step. Then select Colors | Components | Decompose to open a menu, where you can decompose the image, and look for LAB as the color mode (Figure 4). You need to pre-select the option for decomposing the image into layers.

In the next step, activate the a layer from the small image, right click, and select Scale layer to scale it to a higher resolution. Then click on the layer with your mouse and copy it using CTRL + C. Then select the large image and add the layer in the Layers dialog.

Pressing the anchor button at the bottom of the Layers dialog lets you embed the floating selection; you then need to repeat this procedure with the b layer. Finally, use Colors | Components | Recompose to put the color layers back together. The results: An image in a higher resolution with the color information from the smaller image. For comparison, Figure 5 once again shows the grayscale image as a starting point.

Discoloration

The result leaves room for manual rework. For example, the trouser leg is

missing color. The defects take us nicely to the subject of the limitations of this approach. For images with many different colors, colorization does not work, or it only works with restrictions. The software tries to calculate a mean value and assumes that the colors are similar.

Even CNN cannot guess colors. If it discovers a currently unclassified object in a black-and-white photo, say a tent [11], it guesses its color on the basis of the dominant colors – and can get things totally wrong depending on the circumstances.

Having said this, the model achieves “natural-looking” [4] results, according to the study.

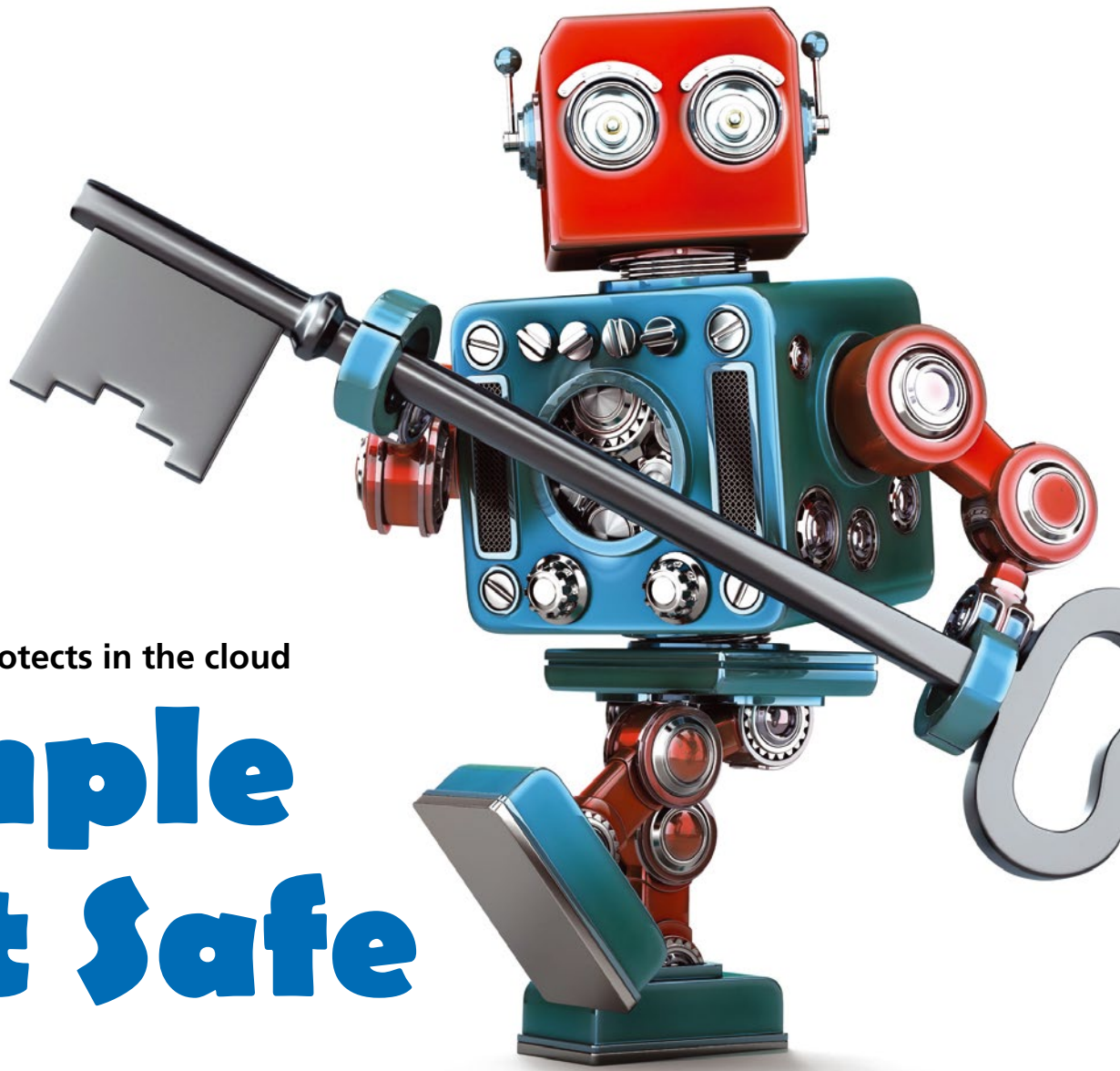
Another useful deep-learning project in the image-processing field is the neural network by the Chinese University of Hong Kong, which knocks out people from photos [12], eliminating the background and leaving the human image only. The procedure demonstrated in the paper is not perfect, but it does achieve better results than the features built into current image-editing programs. The Chinese tool also uses a CNN. ■■■

INFO

- [1] SIGGRAPH 2016: <http://s2016.siggraph.org>
- [2] Github repository for the project: https://github.com/satoshiizuka/siggraph2016_colorization
- [3] Project website: <http://hi.cs.waseda.ac.jp/~iizuka/projects/colorization/en/>
- [4] Scientific work on the project: http://hi.cs.waseda.ac.jp/~iizuka/projects/colorization/data/colorization_sig2016.pdf
- [5] Convolutional neural networks: <https://annalyzin.wordpress.com/2016/01/26/introduction-to-convolutional-neural-network/>
- [6] CNN classifications: <http://cs231n.github.io/classification/>
- [7] Torch: <http://torch.ch>
- [8] Lua: <https://www.lua.org>
- [9] Lab color model: <https://docs.gimp.org/de/glossary.html#glossary-lab>
- [10] Gimp's decompose plug-in: <https://docs.gimp.org/de/plug-in-decompose-registered.html>
- [11] Project presentation: http://hi.cs.waseda.ac.jp/~iizuka/projects/colorization/data/lizukaSIGGRAPH2016_slide.pdf
- [12] Knocking out images with deep learning: http://www.cse.cuhk.edu.hk/leojia/papers/portrait_eg16.pdf



Figure 5: From a grayscale image, the Torch-based colorization software makes a color photo, though not perfect.



Cryptomator protects in the cloud

Simple but Safe

Make files fit for the cloud with Cryptomator by encrypting content and obscuring the name and size of each file. *By Erik Bärwaldt*

Saving files in the cloud is convenient and cost efficient. However, many service providers do not place enough emphasis on data security, allowing content to fall into the hands of unauthorized third parties. Yet, with Linux and the program Cryptomator [1], you can slam the door on snooping.

How It Works

Most cryptographic programs require deep knowledge of encrypting methods and often a great deal of effort when integrating. Cryptomator, on the other hand, is aimed at users looking for a simple and practical approach. The software works transparently in the background, and the dialogs are simple.

The program encrypts data with a 256-bit AES key and a message authentication

code (MAC) master key. Script technology, a method for generating keys that uses a random value and a password to make a dictionary attack more difficult, is used to generate these keys, making brute force attacks difficult. The application comes with a graphical interface, from which you manage the encrypted data that you keep in vaults. From the outset, the software is reminiscent of the command-line program Tomb [2].

Technically speaking, the program functions as a server while encrypting and processing the available data locally on a virtual drive. The program only allows connections on the local system via the loop-back device, a file that provides a virtual block device that does not conform to any hardware on the system and allows you to combine files as a drive. The cryptographic

processing of the individual files is not limited to the content but includes any meta-information and the file's name itself. Additionally, the software changes the size of the file, making it difficult to draw conclusions about the content.

Cryptomator then drops the processed files into the desired vault, which corresponds to the directory that synchronizes with the cloud service. The client for each respective service can then match the encrypted data without the potential need to transfer keys on the server. To use multiple services simultaneously, you will need an independent vault for each cloud service, which you create in the respective sync directory.

If you want to share data with others, they must have access to the relevant vault, know the password, and be able

Lead Image © Kirill Makarov, 123RF.com

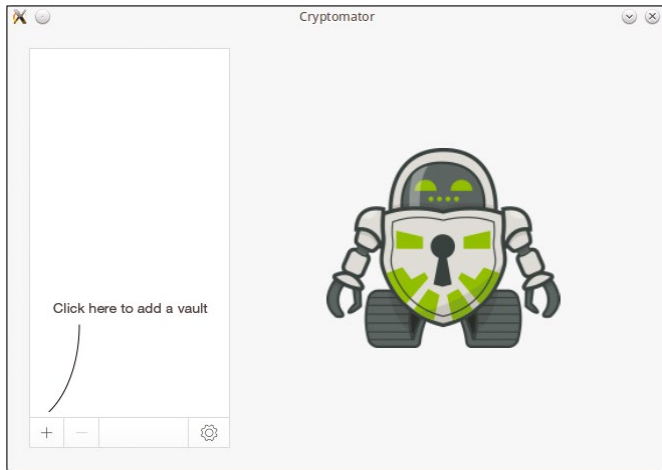


Figure 1: Cryptomator's user interface barely gives you the chance to do anything wrong.

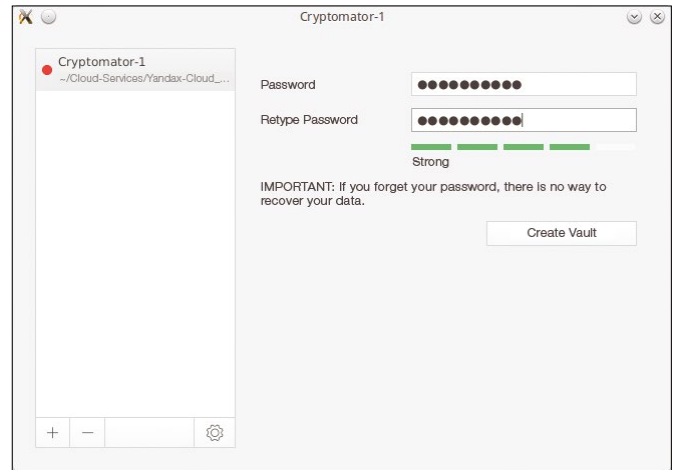


Figure 2: The software indicates whether your password is easy to break.

to send the password securely, such as by an encrypted email. On the other hand, it is not possible to share a single file from a vault with someone. If you have access to the container, you can see everything. If you want to control access in great detail, the only method at your disposal is to create a separate vault for each participant and work with copies of the files.

Unlike container-based programs, Cryptomator only encrypts files that you have changed and currently have loaded. As a result, you can only synchronize modified files. The software works quite quickly, which can pay off in hard cash, particularly in cases of data transfer over mobile devices by UMTS, HSPA, or LTE.

Installation

The Java-based software is available for different distributions on the project's website, where you can get an RPM package and 32- and 64-bit DEB packages. Despite being listed explicitly for Red Hat-based systems, in the test, the packages were also able to run on other distros that use the RPM package management.

Repositories also exist for Ubuntu, and packages for Arch Linux are in the Arch user repository (AUR), which has a collection of scripts that integrate additional software into an Arch installation. A portable version is available for all other systems. In all versions from 1.8 onward, Cryptomator is based on and requires a compatible version of the Java Runtime Environment.

During installation, the program ends up in the `/opt/Cryptomator/app/`

directory; in the *Tools* submenu, you will find a *Cryptomator* entry.

Clients exist not only for Linux, but for Mac OS X and iOS. An Android app is in the works according to the website, and the developers are planning a beta version for fall of this year. If you want to share your data outside the boundaries of the platform, you either need the right system or a measure of patience.

Getting Started

After the program first starts, a window opens; alongside a gear icon for adjusting the WebDAV, the only option it offers is a gear icon for adding a new vault (Figure 1).

Clicking on the plus button and then *Create new vault* in the context menu that pops up opens a file manager, where you create the directory for the encrypted files in the system's cloud folder.

In the next dialog, you set a password for the vault and verify by entering it a second time. The program shows the security of the selected string with a dashed bar colored red or green, depending on the strength of the password (Figure 2). Now your vault is fully ready.

If you click the program window at the bottom right next to the *Lock vault* button on the small triangle, and select the *Reveal drive* option, you can drag and drop the files you want to encrypt into the newly opened file manager window. After storing the files, a graph in the right pane of the program window shows the current throughput in megabytes per second during encryption (Figure 3).

The program stores the encrypted files in the destination folder, at which point the cloud service's original client software typically begins synchronization. Afterward, you can view the number of files saved and the disk space occupied in a conventional file manager like Dolphin (in the Properties dialog for the relevant folder), but not the individual files.

In the cloud service's web interface, you will recognize the individual files,

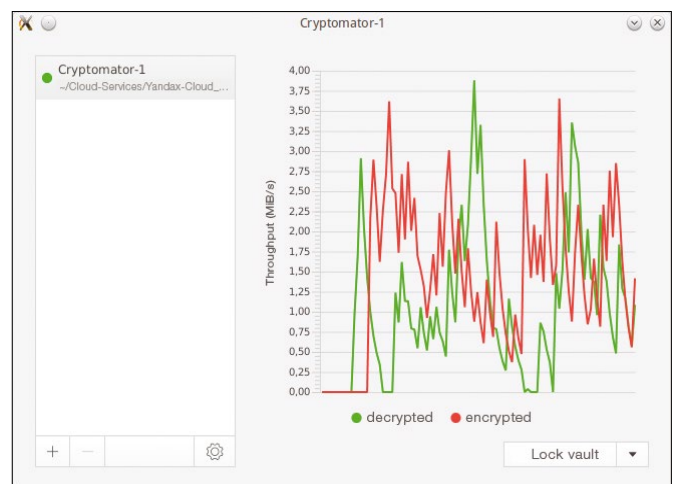


Figure 3: The software displays the throughput it achieves when encrypting the data.

but with obfuscated file names of no significance (Figure 4). You can then download the encrypted files individually from the web interface, although the system identifies them as binary files, which prevents conclusions from being drawn about file types, file names, or file size.

Lock

To exit Cryptomator, you first have to close the current vault by clicking on the *Lock vault* button at the bottom right of the window, at which time the software shows a red dot, instead of green, in front of the vault name in the vault listing in the left pane. Now you can exit the program.

If you close the software without locking all the vaults, the program minimizes without completely shutting down. When launching the vault again, unlock it by entering the password. A short time later, you will again have access to the open virtual drive containing the decrypted files. In subsequent work with files in the virtual directory, the program proceeds to display the throughput when encrypting and decrypting.

Some file managers expect a virtual drive's WebDAV URL, so Cryptomator lets you copy the WebDAV address onto the clipboard. To do this, click

bottom right in the window on the small triangle next to the *Lock vault* button, and then choose the *Copy WebDAV URL* item from the context menu. Now, you can paste the URL from the clipboard into the dialog of the relevant file manager.

If you no longer require a vault, you can delete it. The software allows you to remove the vault completely, including its files, or simply the vault itself. To remove a vault from the list, first dismount it; then, click at the bottom of the list on the minus icon. A query appears suggesting that you only delete the name but not the data from the list.

Now you can delete the vault by clicking on *OK*. Because the virtual filesystem is displayed for this action, you next have to manually delete the encrypted data sets in the system's cloud folder and the cloud service to remove them from the system.

Conclusions

Cryptomator was completely convincing in the test when combined with



different cloud services, working reliably and stably. You can encrypt multiple drives on the intuitive interface, without having to employ a different procedure each

time. Because the keys

stay on the local system,

and the software exclusively encrypts and decrypts the data, it is impossible for third parties to view the content.

Extensive data sets such as multi-gigabyte high-definition video files, however, are not suitable for encrypting with Cryptomator; in such cases, the throughput drops significantly. The tool works quickly on sets of many small files. On fast computers with SSD mass storage, in fact, you see virtually no latency. ■■■

INFO

- [1] Cryptomator: <https://cryptomator.org/de/>
- [2] "Cryptography" by Christoph Langer, *Ubuntu User*, issue 27, 2015, pg. 18, <http://www.ubuntu-user.com/Magazine/Archive/2015/27/Encrypt-your-important-files-using-Tomb>

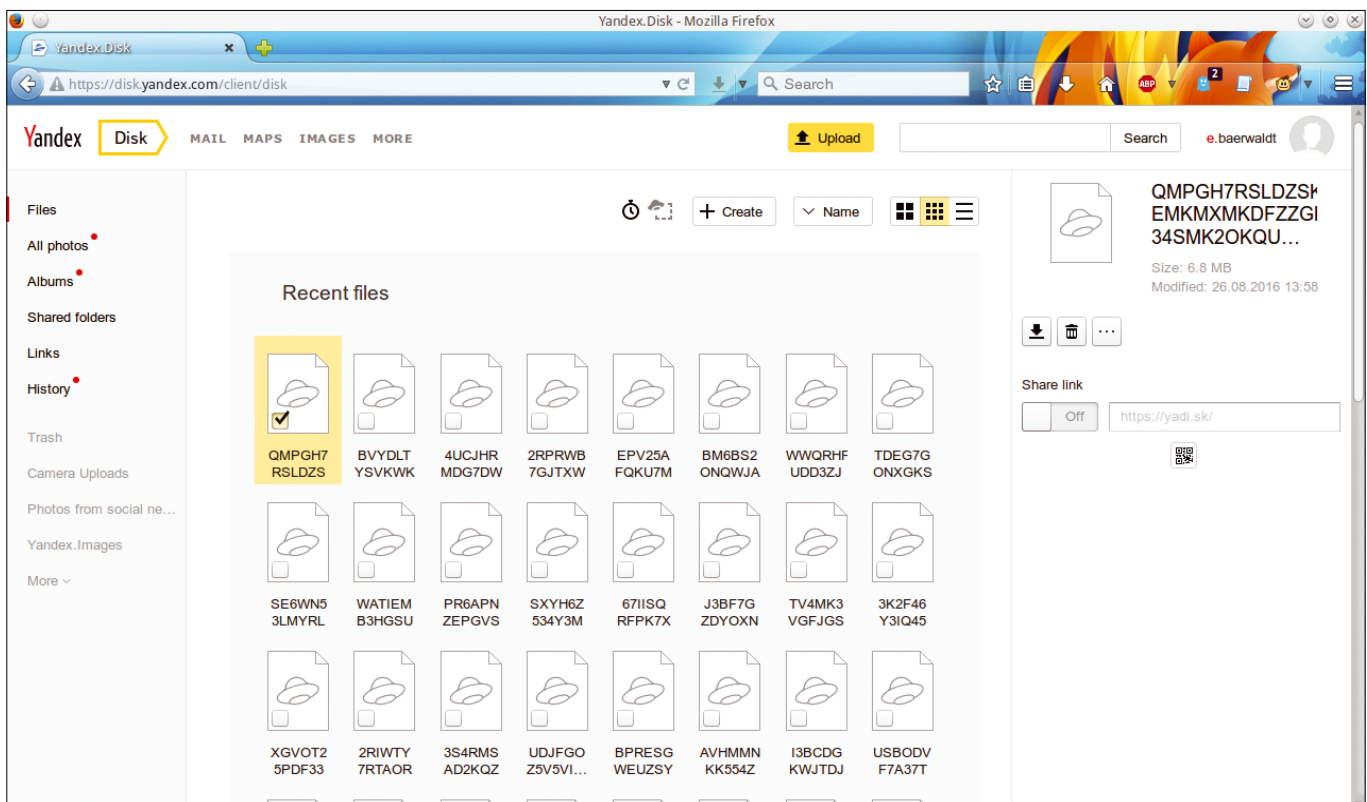
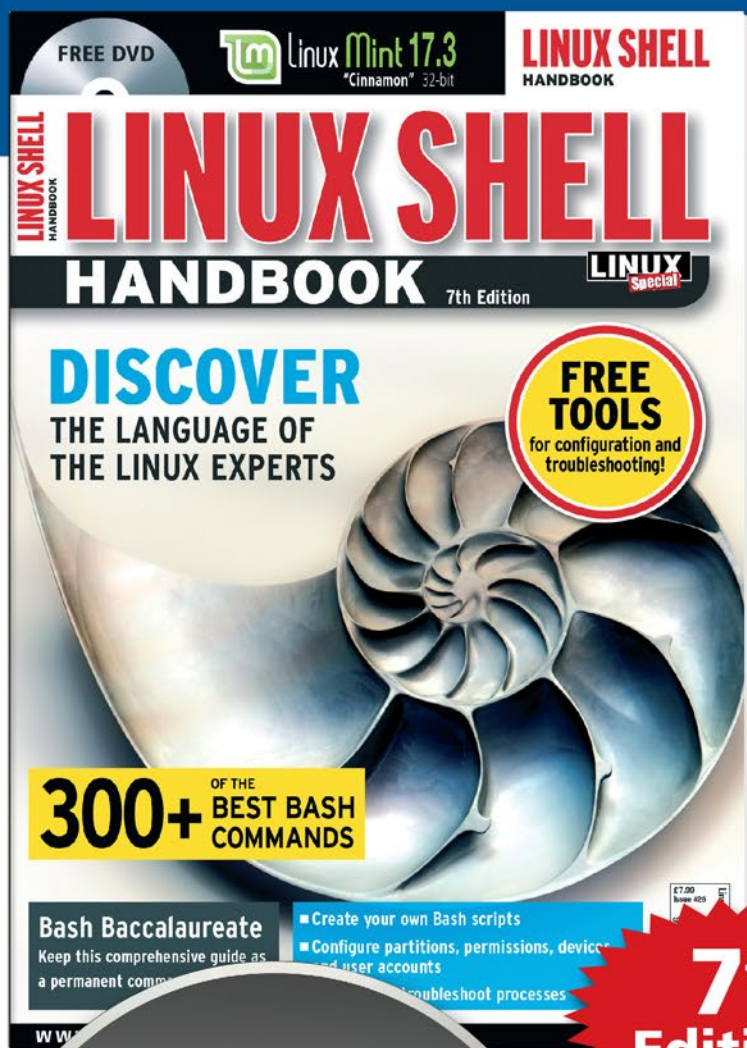


Figure 4: As you can see from the directory of encrypted files, the contents, real names, and actual sizes of these files cannot be identified.

EXPERT TOUCH



Linux professionals stay productive at the Bash command line – and you can too!

The Linux Shell special edition provides hands-on, how-to discussions of more than 300 command-line utilities for networking, troubleshooting, configuring, and managing Linux systems. Let this comprehensive reference be your guide for building a deeper understanding of the Linux shell environment.

You'll learn how to:

- Filter and isolate text
- Install software from the command line
- Monitor and manage processes
- Configure devices, disks, filesystems, and user accounts
- Troubleshoot network connections
- Schedule recurring tasks
- Create simple Bash scripts to save time and extend your environment

The best way to stay in touch with your system is through the fast, versatile, and powerful Bash shell. Keep this handy command reference close to your desk, and learn to work like the experts.



FREE DVD INSIDE!

Linux Mint 17.3
"Cinnamon" 32-bit

ORDER ONLINE:

shop.linuxnewmedia.com/specials

Microsoft's PowerShell for Linux

New Shell on the Beach



Microsoft released its PowerShell under a free license in August 2016 and ported the tool to Linux and Mac OS. Is PowerShell for Linux a mere marketing ploy or a real contender that can compete with native Linux shells? *By Martin Loschwitz*

Microsoft has done its part to perpetuate its image as the bogeyman of the Open Source movement. Who could forget the advertising campaign in which Microsoft gave penguins the mutant heads of other animals – such as a frog with antlers, implying that an open operating system with parts from different places is somehow unnatural? Or when former Microsoft CEO Steve Ballmer called Linux a “cancer” [1]?

For a few years now, however, Microsoft has been toning down its rhetoric and pursuing something more like an appeasement policy, with vague statements about whether Windows would one day have an open license; Microsoft even perplexed the community at one point when it announced it was about to develop its own Linux for its cloud [2].

Recent signs, however, point to genuine improvement in the Microsoft-Linux relationship. A photo from LinuxCon 2016, which made the rounds on social media, showed Linux creator Linus Tor-

valds standing by the Microsoft booth, comfortably sipping a cup of coffee [3]. In 2000, Torvalds made an April Fools joke out of naming Microsoft as a partner for Linux development work. What once was a joke is now reality, and the biggest evidence of Microsoft's commitment to Open Source is the recent release of the Windows PowerShell command environment under a free license, along with the announcement that Microsoft was porting PowerShell to Linux and Mac OS.

What and Why?

The first major PowerShell release appeared in 2007. The idea of PowerShell is to create an alternative to the `cmd.exe` command-line tool, which was included on Windows systems until recently. The `cmd.exe` environment had very limited functionality.

I should also note, for reasons of fairness, that Microsoft considers `cmd.exe` more of a relic from the past than an up-to-date tool for admins. Microsoft introduced PowerShell to provide a tool for the power user. Over the years, PowerShell has gained many fans and indeed acquired prestige.

The goal of PowerShell is to provide a command-line environment that allows the user to execute commands without going through a GUI. Does Microsoft believe it is offering Linux users added

value? What is behind the release of PowerShell for Linux?

The fact that Microsoft wants to promote its .NET framework is likely an important reason for the company's venture into PowerShell for Linux. Part of porting PowerShell to Linux was creating a core .NET platform for the Linux environment. With that platform providing the basis, putting PowerShell on Linux became a comparatively simple exercise.

However, Microsoft is also looking to the cloud. The popularity of Linux instances running in the cloud does not escape Microsoft's watchful eye. Redmond knows its cloud environment will have to support Linux, and by maintaining their own tools for administering both Linux and Windows systems, they can better maintain their position as a player in the server market. The combination of the PowerShell tools, DSC, and the Operations Management Suite makes it possible for you to administer your own instances on Linux in Microsoft's Azure public cloud.

Basic Features

Under the hood, PowerShell is based on Microsoft's .NET framework. It mixes different methods: On the one hand, PowerShell allows you to execute simple command-line commands, just as admins would be familiar with from Linux. Different commands can also be combined to produce typical shell scripts.

AUTHOR

Martin Gerhard Loschwitz works as the OpenStack Team Leader at Sys Eleven in Berlin. In his free time, he enjoys sailing, preferably on the Baltic Sea.



PowerShell controls central shell functions on Linux as well, including the ability to redirect the output of a program to another by piping. On the other hand, PowerShell follows an object-oriented approach: You can define methods and objects that then use the methods.

The central pivot points for working with PowerShell are the commandlets. In PowerShell speak, a cmdlet is a .NET class that performs a specified task after launching. Cmdlets are essentially the actual commands, which can be bundled together in the form of scripts or even binaries. Microsoft has invested a lot of work in equipping PowerShell with many useful cmdlets. Straight after installation, there are many commands available, such as for deleting files or directories, creating folders, or displaying the folder contents (Figure 1).

According to the PowerShell standard, cmdlets must have a unique name consisting of a verb and object. As a result, the cmdlet for deleting folders or files is officially `remove-item`. Another example: The cmdlet for downloads from the Internet is `invoke-WebRequest`; if you are accustomed to Linux, an alternative for `invoke-WebRequest` would be `wget` or `curl`.

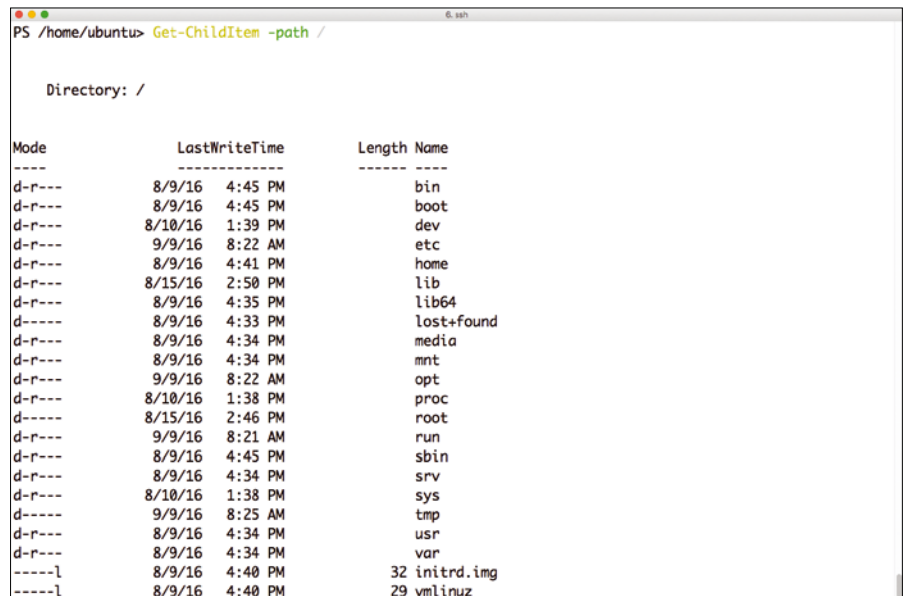
To make the transition to PowerShell easier for experienced Linux admins, Microsoft alias-mapped common Linux commands to PowerShell cmdlets. After complaints from the user base, however, Microsoft rejected this approach, because it ultimately prevents users from executing the real command-line programs, such as `ls` or `cd`.

Installation on Linux

Microsoft is clearly keen to keep entry barriers as low as possible for PowerShell on Linux. The company is making the source code for PowerShell Version 6 available to the public on GitHub, but it is also offering a package version for the major Linux distributions. So it is very easy to get started, for instance, on Ubuntu 16.04 [4].

After the installation of the `libunwind8` and `libc6u55` packages, you can install the PowerShell package with `dpkg -i` (Figure 2).

The service would be even better if PowerShell came from a real package repository with signed release files, though this may still be too great an effort for Microsoft. Because packages for various



```
PS /home/ubuntu> Get-ChildItem -path /

Directory: /

Mode                LastWriteTime         Length Name
----                -
d-r---             8/9/16  4:45 PM             bin
d-r---             8/9/16  4:45 PM             boot
d-r---             8/10/16  1:39 PM             dev
d-r---             9/9/16  8:22 AM             etc
d-r---             8/9/16  4:41 PM             home
d-r---             8/15/16  2:50 PM             lib
d-r---             8/9/16  4:35 PM             lib64
d-----           8/9/16  4:33 PM             lost+found
d-r---             8/9/16  4:34 PM             media
d-r---             8/9/16  4:34 PM             mnt
d-r---             9/9/16  8:22 AM             opt
d-r---             8/10/16  1:38 PM             proc
d-----           8/15/16  2:46 PM             root
d-r---             9/9/16  8:21 AM             run
d-r---             8/9/16  4:45 PM             sbin
d-r---             8/9/16  4:34 PM             srv
d-r---             8/10/16  1:38 PM             sys
d-----           9/9/16  8:25 AM             tmp
d-r---             8/9/16  4:34 PM             usr
d-r---             8/9/16  4:34 PM             var
-----l           8/9/16  4:40 PM             32 initrd.img
-----l           8/9/16  4:40 PM             29 vmlinuz
```

Figure 1: The classic `Get-Childitem` PowerShell cmdlet is equivalent to the `dir` command used with `cmd.exe`.

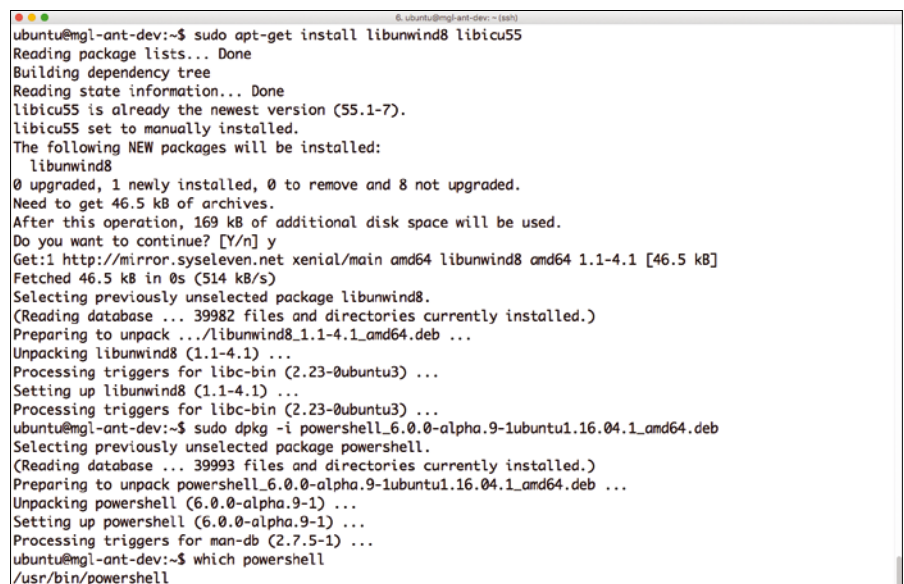
RPM-based distributions are also available from the provider, Microsoft would have to convert the repository for several package systems. Even without their own repository, it works: Good job! So all kinds of others software projects can take a big page out of Microsoft's book.

.NET Core: Slimmed-Down .NET

In PowerShell's documentation for Linux and Mac OS, the Microsoft developers point out that Windows admins on Linux will not find all of the usual PowerShell commands. PowerShell for Linux is not based directly on the large .NET frame-

work, but on a slimmed-down version called .NET Core. For example, the PowerShell functions referring to the Windows interface do not exist on Linux systems or Mac OS computers. Similarly, several cmdlets that Windows admins will be familiar with from their usual environment are absent from PowerShell for Linux.

The slimmed-down .NET version is also important in another respect: If you find cmdlets or scripts for PowerShell on the Internet, you can generally expect the Windows PowerShell to run the script correctly. Ultimately, cmdlets are simply .NET classes that include other, commonly available .NET classes. If a



```
ubuntu@mg1-ant-dev:~$ sudo apt-get install libunwind8 libc6u55
Reading package lists... Done
Building dependency tree
Reading state information... Done
libc6u55 is already the newest version (55.1-7).
libc6u55 set to manually installed.
The following NEW packages will be installed:
  libunwind8
0 upgraded, 1 newly installed, 0 to remove and 8 not upgraded.
Need to get 46.5 kB of archives.
After this operation, 169 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://mirror.sysseven.net xenial/main amd64 libunwind8 amd64 1.1-4.1 [46.5 kB]
Fetched 46.5 kB in 0s (514 kB/s)
Selecting previously unselected package libunwind8.
(Reading database ... 39982 files and directories currently installed.)
Preparing to unpack .../libunwind8.1.1-4.1.amd64.deb ...
Unpacking libunwind8 (1.1-4.1) ...
Processing triggers for libc-bin (2.23-0ubuntu3) ...
Setting up libunwind8 (1.1-4.1) ...
Processing triggers for libc-bin (2.23-0ubuntu3) ...
ubuntu@mg1-ant-dev:~$ sudo dpkg -i powershell_6.0.0-alpha.9-1ubuntu1.16.04.1.amd64.deb
Selecting previously unselected package powershell.
(Reading database ... 39993 files and directories currently installed.)
Preparing to unpack powershell_6.0.0-alpha.9-1ubuntu1.16.04.1.amd64.deb ...
Unpacking powershell (6.0.0-alpha.9-1) ...
Setting up powershell (6.0.0-alpha.9-1) ...
Processing triggers for man-db (2.7.5-1) ...
ubuntu@mg1-ant-dev:~$ which powershell
/usr/bin/powershell
```

Figure 2: After the installation of `libunwind8` and `libc6u55`, PowerShell can install with the aid of the .deb product bundle from Microsoft.

class referenced in the script is missing from the .NET Core collection, it won't run in PowerShell for Linux.

OMI – The Rebirth of WMI

Another component that PowerShell relies on in the background, and that is lacking on Linux systems, is Windows Management Instrumentation (WMI). WMI performs important computer-management tasks on Windows systems. Windows hosts can be automatically administrated via WMI, for instance, by rolling out the code on the target system.

WMI is Microsoft's method for realizing the principle of Web-Based Enterprise Management (WBEM) and implementing an interface for the Common Information Model (CIM) according to DMTF standards. PowerShell's open source properties cease when it comes to the topic of WMI. The PowerShell extensions related to WMI are not in the open source part of PowerShell and therefore don't work in the Linux version.

Microsoft is promising to offer a solution for the lack of WMI support; the company currently refers users to DSC for Linux, which has been available for quite a while now. The idea behind DSC is to compensate for the lack of WMI functionality with OMI. OMI, the Open Management Infrastructure, is an alternative to WMI by the Open Group, and it is available online as a free download. DSC lets PowerShell execute tasks on Linux similar to those performed on Windows via WMI.

DSC: Desired State Configuration

The component within PowerShell that prepares other computers using a preset configuration is called Desired State Configuration (DSC). Microsoft had already released a limited version of DSC for Linux in 2015. Like Puppet, DSC implements declaratively-formulated settings alongside the admin instructions on the host. Another similarity with Puppet is that DSC is idempotent, meaning that it delivers the same result with multiple runs as with one simple run.

An interesting fact about DSC is that it also works similarly to Puppet under the hood. You supply the desired configuration in the form of a config document (MOF file), then transfer it from its PowerShell environment to the Linux ma-

chine via a push command. The PowerShell documentation on DSC demonstrates a solid example for creating a file in /tmp on the target host.

However, that naturally has little to do with the practical reality. While WMI is adjusted for the system on Windows hosts, the OMI-based approach currently remains more or less useless in its effect for Linux and Mac OS. It remains to be seen whether Microsoft is looking to develop DSC for Linux into a proper management tool, or whether in the end the field will be left to established solutions such as Puppet.

As previously mentioned, DSC for Linux was originally a low-priority effort from Microsoft, with the company implementing the necessary PowerShell commands as Python scripts. With the publication of PowerShell for Linux, they have avoided such a necessity. For the foreseeable future, Microsoft is likely to continue to offer DSC on Linux from the real PowerShell.

With the exception of DSC, little is going on with PowerShell in its first version for Linux. WinRM functionality is not present, which means it is not possible to use PowerShell to send a computer remote commands – a PowerShell equivalent to SSH functionality is simply not available for Linux. Microsoft has at least recognized this problem and has promised to address it. However, it is impossible to predict how the implementation will look in practice. Microsoft is presumably relying on established protocols such

as OpenSSH for the solution. Altogether, PowerShell is likely to develop in the future into a universal tool aimed at Windows and Linux systems equally.

Scripting

Shell scripts are still a hotly disputed Linux topic: Some users love the ability to bundle commands together in a file and then execute them. Others see the work of the devil in shell scripts and argue that any shell script longer than ten lines should be written in a scripting language such as Perl or Python. Those who advocate shell scripts can be happy with PowerShell, which offers the ability to execute commands via a PowerShell script. The principle works the same as with typical Linux shells. It is important that the script file ends with .ps1.

As with Bash, you string together the cmdlets you want to call in PowerShell scripts. PowerShell scripts can also be equipped with conditions, so such as If, Elseif, and Else (Figure 3).

You will also find an option to store your own functions in the form of modules: The example of Apache [5], which the PowerShell developers list in their GitHub directory, explains this feature well. As admin, you can define functions with the keyword Function in a file with a name that ends with .psm1. As is customary with other shells, functions can have parameters. The module definition then loads each shell script with the Import-Module cmdlet; your own definitions are already available within the shell script.

```
#Region utility functions

$global:sudocmd = "sudo"

Function GetApacheCmd{
    if (Test-Path "/usr/sbin/apache2ctl"){
        $cmd = "/usr/sbin/apache2ctl"
    }elseif (Test-Path "/usr/sbin/httpd"){
        $cmd = "/usr/sbin/httpd"
    }else{
        Write-Error "Unable to find httpd or apache2ctl program. Unable to continue"
        exit -1
    }
    $cmd
}

Function GetApacheVHostDir{
    if (Test-Path "/etc/httpd/conf.d"){
        Return "/etc/httpd/conf.d/"
    }
    if (Test-Path "/etc/apache2/sites-enabled"){
        Return "/etc/apache2/sites-enabled"
    }
}

Function CleanInputString([string]$inputStr){
    $outputStr = $inputStr.Trim().Replace('\n','').Replace('\n','')
    $outputStr
}

"Downloads/Apache.psm1" 234L, 8221C
```

Figure 3: PowerShell supports conditional expressions.

An Alternative?

It is probably beyond dispute that Microsoft's has great ambitions for open source, since they have invested substantial resources into making an open source product from PowerShell. Yet PowerShell for Linux currently exists as something more like a proof of concept.

In the GitHub directory of PowerShell, Microsoft maintains a list of problems that currently rattle PowerShell on Linux. On the one hand, publicly tracking the problems is laudable, since it shows a commitment to openness. On the other hand, the list reads like a message from the house of horrors, with some basic functions still defective at the time of this article went to press.

For example, admins are accustomed to using wildcards from Linux shells so they can select several files dynamically on the command line (* for multiple characters or ? for one character). Wildcards work on PowerShell, but only for cmdlets. If you call a native Linux binary from PowerShell and pass a wildcard, you will receive an error because PowerShell does not automatically replace the wildcard character. (Figure 4).

Calling native Linux binaries on PowerShell has yet more unpleasant side effects: The return value of cmdlets is always an object. PowerShell relies on objects to facilitate features such as forwarding the output of a program to other programs as input. Native Linux programs produce the classic return values and no objects, on the other

hand, meaning that the piping does not work here.

The fact that PowerShell does not distinguish between upper- and lowercase by design is at least as serious. In particular, this affects finished PowerShell scripts from the web: If you download a finished script for a particular task, you will have to hope that its author uses exactly the same name the current module has in the filesystem when loading external modules, for instance. In a Linux environment, you would simply receive an error message instead.

The topic of case sensitivity is also a problem for Windows admins arriving at Linux from PowerShell. Upon completing the input with the [Tab] key, PowerShell displays the relevant program on Windows, no matter whether it is written with matching upper- and lowercase. If you are on Linux, however, and enter MK in the belief that it will auto-complete to mkdir, you will simply get another error message.

To top it all, some stumbling blocks will arise from fondly-held conventions if Linux admins use PowerShell, or Windows PowerShell users use Linux. It is possible to stop or start programs via PowerShell on Windows, for instance, but the commands necessary for this do not work on Linux.

If you are not scared off by these teething troubles and attempt your first steps with PowerShell, Microsoft offers very generous help. Along with PowerShell for Linux, the giant from Redmond has provided complete examples of new

cmdlets and demonstrated how you can use scripting languages (such as Python) in a sensible way with PowerShell. If you are interested, you can find the full sample archive online [6].

Conclusion

At the time of this test, PowerShell for Linux is still a construction site, which will hardly put experienced Bash admins at ease. However, Microsoft is still calling PowerShell for Linux an alpha version, and public development (along with bug searching) has only been underway for a few months.

Much more important than the state of the product is the significance of PowerShell for Linux as a signal that Microsoft no longer views Linux as a nemesis, but as a competitor, and even a genuine partner for possible cooperation. Microsoft CEO Satya Nadella was photographed a few months ago with a PowerPoint slide that read "Microsoft Loves Linux," causing some amusement on the scene. The release of Linux PowerShell and the work that Microsoft has been putting into in the product prove that those were not empty words.

PowerShell for Linux is only at the beginning of its development. If Microsoft maintains its current pace, PowerShell could become a serious competitor for established Linux shells within the foreseeable future. ■■■

INFO

- [1] Steve Ballmer, Linux is a Cancer: http://www.theregister.co.uk/2001/06/02/ballmer_linux_is_a_cancer/
- [2] Linux from Microsoft for the cloud: <http://azure.github.io/SONiC/>
- [3] Torvalds at the Microsoft stand: <http://fosshbytes.com/wp-content/uploads/2016/08/linus-torvalds-meets-microsoft-guys-linuxcon.jpg>
- [4] PowerShell for Ubuntu 16.04: https://github.com/PowerShell/PowerShell/releases/download/v6.0.0-alpha.9/powershell_6.0.0-alpha.9-1ubuntu1.16.04.1_amd64.deb
- [5] Apache Script for PowerShell: <https://github.com/PowerShell/PowerShell/tree/master/demos/Apache>
- [6] Demos for PowerShell: <https://github.com/PowerShell/PowerShell/tree/master/demos>

```
PS /home/ubuntu> dir

Directory: /home/ubuntu

Mode                LastWriteTime         Length Name
----                -
d-----           9/5/16 12:58 PM             consul
d-----           8/16/16 11:03 AM             flask
d-----           9/2/16  1:25 PM             glance
d-----           9/6/16  1:13 PM             heat
d-----           9/7/16 10:03 AM             horizon
d-----           9/6/16  1:16 PM             nova
d-----           9/6/16  1:12 PM             os-client-config
d-----           9/6/16  1:00 PM             shade
d-----           8/16/16 10:54 AM             six
d-----           9/5/16  1:20 PM             storagemgr
-----           8/16/16 11:39 PM    40928824 powershell_6.0.0-alpha.9-1ubuntu1.16.04.1_amd64.deb

PS /home/ubuntu> ls
consul glance horizon os-client-config shade storagemgr
flask heat nova powershell_6.0.0-alpha.9-1ubuntu1.16.04.1_amd64.deb six
PS /home/ubuntu> ls *
/bin/ls: cannot access '*': No such file or directory
PS /home/ubuntu>
```

Figure 4: The dir and ls commands call up different programs, although globbing with wildcards does not work with ls.



Exploring the FOSS alternative to Twitter

Free and Social

GNU social is a free and open source microblogging platform similar to Twitter. You can join one of the public federated servers or set up your own in minutes. *By Nate Drake*

In July of this year, Milo Yiannopoulos was banned from Twitter permanently as the website cracked down on several alleged incidents of racial abuse against *Ghostbusters* actor Leslie Jones [1]. The decision has sparked a wider discussion about freedom of expression online, and Twitter hasn't helped the situation by applying its own policies inconsistently at times. For example, *Los Angeles Times* journalist Fredrik deBoer pointed out that members of the Somali Al-Shabab militia have seemingly been permitted to discuss acts of violence openly despite being flagged by users [2].

The debate of freedom of expression versus toleration of hate speech will not be solved here. However, it's apparent that as long as people use public social networks like Twitter, they are not in control of their data. A social network giant can censor

your messages or even sell your public tweets to data miners, as Twitter is reported to have done by *The Guardian* [3].

Enter the GNU

GNU social is quite simply a decentralized microblogging platform (Figure 1). It

can be installed on a private server and can boast that it is both sponsored by and used by the Free Software Foundation [4].

Crucially, GNU social allows for open and distributed communications between microblogging communities. Services are federated, so although a

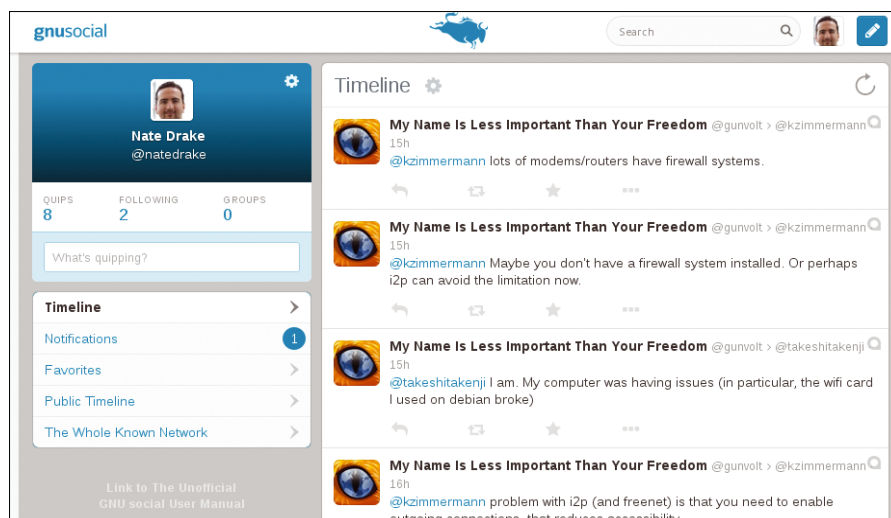


Figure 1: The author's own GNU social account provided by quitter.no.

AUTHOR

Nate Drake is a freelance journalist specializing in cybersecurity and retro tech.

business or individual remains in control of their data, any user of any instance can follow, be followed by, and communicate with users of any other instance [5]. To follow a remote user, simply combine their username with the name of their chosen GNU node (e.g., janedoe@quitter.is).

The decentralized nature of GNU social naturally works both ways. If an overzealous admin censors users' online activity or is forced to do so via oppressive laws, users can simply create a new account on another GNU social node.

Given that the software can be deployed in minutes, it exists on hundreds of machines. A link to a full list of independent GNU social instances is available via the GNU main page [6]. For those who do not have their own server, there are several public deployments including quitter.se, quitter.no, quitter.is, quitter.es, and gnosocial.no. The Free Software Foundation itself runs a GNU social node at <https://status.fsf.org/>.

Quitter.es (Quitter España), for example, was created last year in a bid to cope with a huge migration of Spanish Twitter users to GNU social. On January 14, user @Barbijaputa – who was known for being openly critical of the Spanish government – was suspended from Twitter. The following day she created a profile on quitter.se and a number of her 167,000 followers chose to migrate with her [7].

StatusNet and GNU

GNU social has something of a convoluted history. It was started in 2010 as a spin off of the GNU FM project, which was designed to power music communities such as Libre.fm. In 2013, Evan Prodromou, who had recently discontinued work on his StatusNet project, kindly provided a copy of his work to be merged into the Free Software Foundation's own project [8].

Prodromou abandoned his work on StatusNet to concentrate on a federated social networking protocol known as pump.io, which is used in web applications like GNU MediaGoblin [9]. Today, GNU social has been installed on both public and private servers, although the decentralized nature of the project makes it difficult to calculate exact numbers of individual users.

Getting Started with GNU Social

The first step to use GNU social is to choose a server. You can host a server yourself or use one of the more popular public implementations like Quitter or LoadAverage. Quitter's interface is most similar to Twitter, so it provides the easiest transition. Moreover quitter.no and quitter.is are based in Norway and Iceland, respectively, which are not regulated by the European Data Protection Directive, which allows sharing of private data within EU states under certain circumstances [10]. Additionally, both sites are available in English and a range of European languages, as well as Arabic, Persian, Chinese, and Japanese.

Once you've registered your account on either a public server or your own GNU social Node, there are a range of clients to choose from. This is usually the stage where many users are unable to proceed, because typing quitter or loadaverage into their app store of choice will not yield anything meaningful. A fairly comprehensive list of all available clients is available online [11], but Android and iOS users will be pleased to hear that there are mobile versions of all the instances of Quitter.

Android users may wish to install the Mustard client (Figure 2), which has additional features like attachments, bookmarks, filters, and geolocation [12]. Users of iOS devices can download Mayo for Status.net free of charge to log in to their own instance of GNU social. In both cases, when asked to provide the server name, it's important to include the full link (e.g., <https://quitter.se>).

Users of desktop machines can simply visit the GNU social instance in their browser. Alternatively, users of KDE can install Choqok (from the ancient Persian word for sparrow), which supports both Twitter and GNU social, making it easy to flit between the two [13]. Heybuddy is also an excellent Python client for GNU social, which will run in any desktop environment [14].

Once the necessary plugins have been installed, your GNU social instance may also be compatible with XMPP clients. The specific steps to get your XMPP client to talk to your GNU social users will vary from application to application but the Skilled Tests wiki

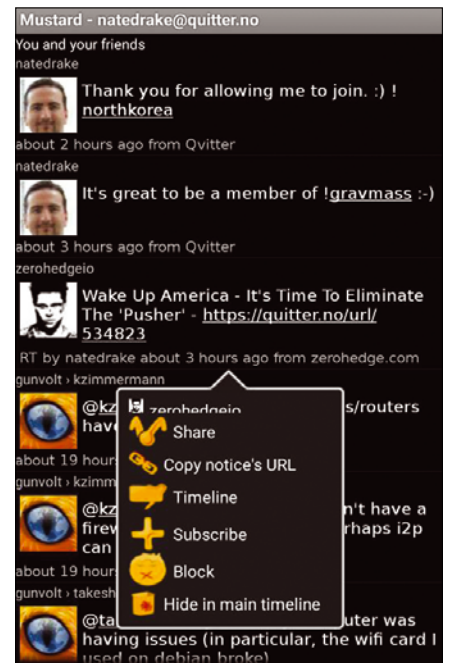


Figure 2: Those familiar with the Twitter mobile app will find Mustard easy to master.

has an excellent walk-through of the basic steps [15].

GNU Groups

Despite the strong similarity to Twitter, GNU social has an excellent feature of its own in that you can set up and join groups. These groups can span across servers, which makes them useful to keep conversations in sync. The specific steps for joining groups will vary depending on your GNU social node (Figure 3). On Quitter or LoadAverage, the easiest way is to search for the group name using the search bar at the top right.

One example is Richard Stallman's group, which celebrates the imaginary holiday of Grav-Mass. The group shares scientific knowledge as well as discussing experiments that could answer unresolved questions about the world [16]. A search of all public postings for "gravmass" will reveal some posts with the group name preceded by an exclamation mark (!). Click on any of these to be taken to the group's main page and subscribe. You will be returned to your own GNU social node after doing to confirm you want to join the group.

Alternatively, if you know the URL for the group in question, click *Join remotely* --if your GNU social ID is not on that instance – then enter the ID itself.

gnusocial New Quitter.no FAQ Settings

Recent tags
People
Popular

GROUPS
northkorea
gravmass

Nickname:
1-64 lowercase letters or numbers, no punctuation or spaces.

Full name:

Homepage:
URL of the homepage or blog of the group or topic.

Description:
Describe the group or topic in 1000 characters or less.

Location:
Location for the group, if any, like "City, State (or Region), Country".

Aliases:
Extra nicknames for the group, separated with commas or spaces. Maximum 3 aliases allowed.

Figure 3: You can enter up to three aliases to allow people to find your group. Currently, this only works for users on the same GNU social node as the group itself.

As before you'll be taken back to your own GNU social Instance to confirm that you wish to subscribe by clicking *Join*.

If the current list of federated GNU social groups [17] doesn't meet your needs, then new groups can be created by visiting <http://yourgnusocialinstance.com/group/new> (e.g., <http://quitter.no/group/new>).

You'll be asked on the web form to provide a nickname for the group all in lowercase letters (e.g., travel-lovers) as well as a homepage and description. You also have the option to set aliases for the group (e.g., the country Myanmar is sometimes referred to as Burma, so you might want both *!myanmar* and *!burma* to point to the same group).

they are using the GNU social node where the group is located. Once your group has been created, you may wish to post the link in *!fedgroups* to include it in the Skilled Tests wiki list.

Whether you join an existing group or create your own, in order to post, you simply include the group name or an alias in your message (e.g., *!burma*). If the group is private, then only other members will see it; otherwise, the post will be visible to those who visit the group page also.

GNU Social Server

For users with access to their own server, the requirements for installing GNU social are not too onerous.

Finally, you have an option to make the group Private. If this is checked then members must be approved by an admin and messages remain private.

Make sure to provide the URL of the group to allow others to join. If you have chosen an alias, then the URL will still direct visitors to your group provided

Full instructions for installation are available on GNU social's GitHub page [18], but, in brief, prerequisites include PHP 5.5+ and MariaDB 5+ /MySQL 5.5+ . A web server such as Apache, lighttpd, or Nginx is also required (Figure 4).

The official website recommends downloading a tarball of the latest version of GNU social onto your web server and unpacking it, but it is probably easier simply to use *git* to clone the contents of the source code from the website [19] into a local folder.

After modifying permissions to allow users to access the *gnu-social* folder, the next step is to use *mysqladmin* to create a database and root password specifically for GNU social. Once this has been done and a new database account has been created, you should be able to visit <https://yourserveraddress.com/install.php> to create an admin account and register new users.

At this stage, you may wish to follow the steps in the installation manual to allow fancy URLs for registering new accounts or install the SphinxSearch plugin to allow searching for other users or images.

GNU Grumbles

The chief advantages of GNU social have already been covered in that it provides for greater freedom of expression, potentially more privacy, and allows you to retain your user data.

The easiest way to get started with GNU social is to use a public instance.

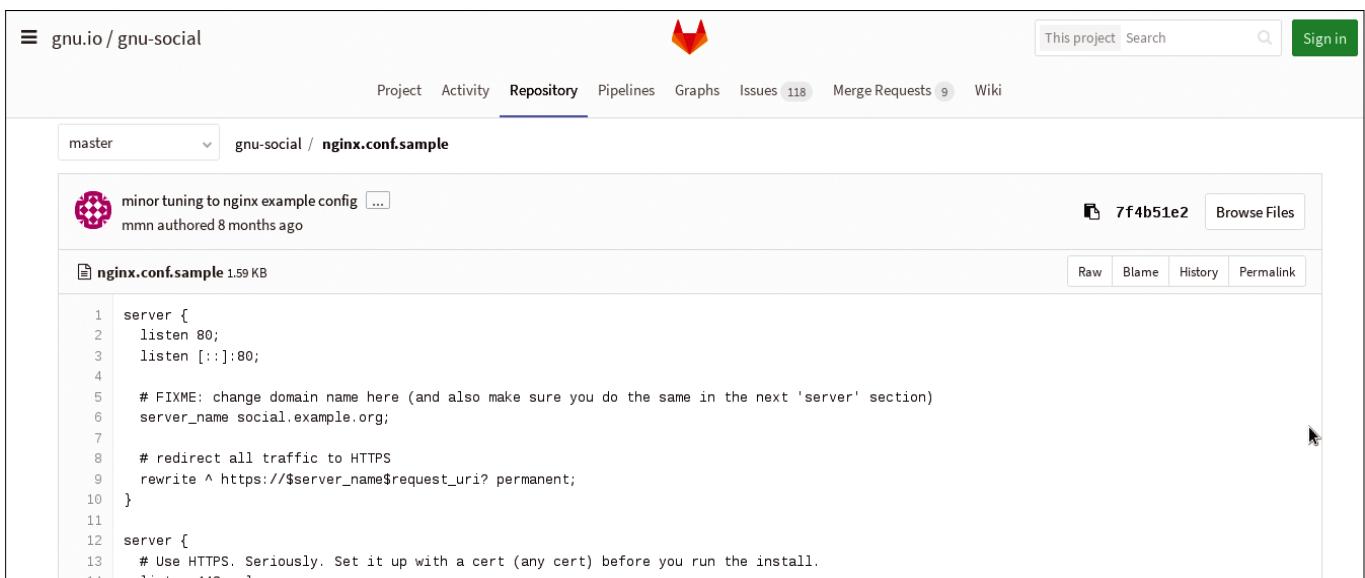


Figure 4: Setup is quick and easy. The GitHub page contains sample configuration files for common servers such as Nginx.

However, admins can and do moderate public posts according to their own policies. Quitter.no's user policy, for example, states:

"Quitter.no is not a service and you are not a customer here. We are a small part of a bigger social change, creating a large decentralized community. This means that we don't have to be neutral toward the content on our GNU social instance. If you don't like the direction this instance is going, you are free to move to another instance or start your own."

The website also states in bold that users guilty of perceived racism, sexism, ableism, transphobia, and so on will be removed (Figure 5). Note that even if the site admins believe passionately in freedom of speech, the server may also be located in a jurisdiction which restricts it, so consider the location carefully.

Quitter also reminds users that they are free to create their own GNU social node, so their messages and accounts are protected from arbitrary deletion. Sadly, there's no built-in method to migrate existing posts and user data from one instance to another. The installation manual recommends using `mysqldump` to back up the database and `tar` to back up the web directory.

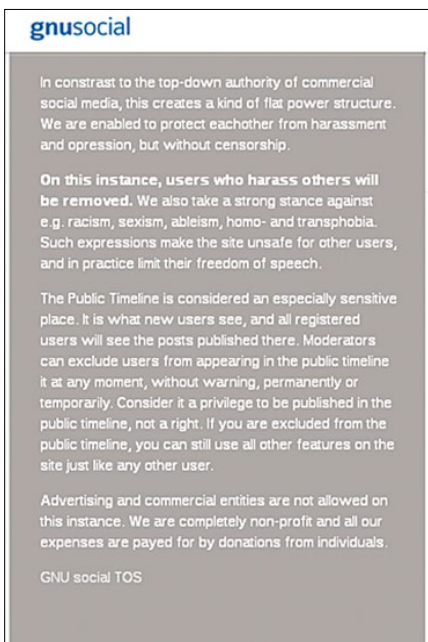


Figure 5: Quitter.no reserves the right to restrict anything it sees as discriminatory. Advertisements of any kind are also not allowed. If you want to control all content, you can set up your own GNU social instance.

One advantage of a centralized social media network like Twitter is that when a public post is removed, it's gone from the site for good. If a user on a GNU social node removes a post from their public timeline, it will be removed from their own instance but copies may remain on other federated servers. For this reason, if there's no need for your posts to be public (e.g., you are using GNU social to collaborate with colleagues on a project), you can make your node private so it doesn't federate with other servers [20].

Migrating from Twitter to GNU social is a painless process, but it may be difficult to persuade friends and family to do the same. As such, you may wish to link your GNU social account with Twitter.

GNU social claims to be compatible with Twitter's API. This theoretically means you can cross post to Twitter although specific steps are not available on the GNU social main page. However, users who register with <https://gnusocial.net> can use the

settings menu on the main site to authorize their account to post to Twitter [21]. Naturally, any posts made to Twitter will need to be done from a Twitter account, which can be suspended or deleted.

If you are running your own GNU social instance, the TwitterBridge plugin not only allows posting notices to other Twitter accounts but also allows users to authenticate using Twitter, as well import your Twitter friends' tweets [22]. You'll also need to create a Twitter App for GNU social to access your account at <https://apps.twitter.com/>.

Overall, GNU social offers a crisp alternative to Twitter and has the power to safeguard freedom of expression online. Despite the lack of some of the more advanced posting features of Twitter, such as URL lookups and meta tags, the project has promise. Thanks to Bytemark Data Centres [23], the GNU social Project has found a new home in the UK, and it's likely that more public and private servers will be created in the near future. ■■■

INFO

- [1] *Washington Post* article: <https://www.washingtonpost.com/news/the-intersect/wp/2016/07/21/what-it-takes-to-get-banned-from-twitter/>
- [2] *Los Angeles Times* Op-Ed: <http://www.latimes.com/opinion/op-ed/la-oe-deboer-yanniapoulos-jones-twitter-speech-20160722-snap-story.html>
- [3] *The Guardian*: <https://www.theguardian.com/technology/2015/mar/18/twitter-puts-trillions-tweets-for-sale-data-miners>
- [4] GNU social: <https://gnu.io/social/>
- [5] Vinilox's GNU social FAQ: <https://status.vinilox.eu/doc/faq#faq-2>
- [6] Independent GNU social instances: http://skilledtests.com/wiki/List_of_Independent_GNU_social_Instances
- [7] Free Software Foundation: <https://www.fsf.org/blogs/community/thousands-of-spaniards-leave-twitter-for-gnu-social>
- [8] GNU social: <https://gnu.io/social/>
- [9] MediaGoblin: <http://mediagoblin.org/>
- [10] Linux Veda: <http://www.linuxveda.com/2015/06/05/gnu-social-vs-twitter/>
- [11] GNU social clients: http://www.skilledtests.com/wiki/GNU_social_clients
- [12] Mustard: <https://play.google.com/store/apps/details?id=org.mustard.android&hl=en>
- [13] Choqok: <http://choqok.gnufolks.org/>
- [14] Heybuddy: <http://www.jezra.net/projects/heybuddy>
- [15] Statusnet problems: http://www.skilledtests.com/wiki/Statusnet_problems#use_XMPP
- [16] Celebrate Grav-Mass: <https://stallman.org/grav-mass.html>
- [17] List of federated GNU social groups: http://www.skilledtests.com/wiki/List_of_federated_GNU_social_groups#how_to_create.2Fjoin_a_group_in_GNU_social.3F
- [18] GNU social installation: <https://github.com/foocorp/gnu-social/blob/master/INSTALL>
- [19] Clone source code: <https://git.gnu.io/gnu/gnu-social.git>
- [20] Make a private node: <https://git.gnu.io/gnu/gnu-social.git>
- [21] GNUsocial.net FAQ: <https://gnusocial.net/doc/faq#faq-12>
- [22] GNU social plugins: <https://github.com/foocorp/gnu-social/tree/master/plugins/TwitterBridge>
- [23] GNU social: <https://gnu.io/social/>

Klaus Knopper answers your Linux questions

Ask Klaus!

By Klaus Knopper



KLAUS KNOPPER

Klaus Knopper is an engineer, creator of Knoppix, and co-founder of LinuxTag expo. He works as a regular professor at the University of Applied Sciences, Kaiserslautern, Germany. If you have a configuration problem, or if you just want to learn more about how Linux works, send your questions to: klaus@linux-magazine.com.



Good Morning Klaus,

I am using Knoppix 7.7 obtained from *Linux Magazine*. I boot from a boot-only CD to an 8GB memory stick with an encrypted Reiser partition. The OS remembers my trash can on the desktop settings, personalized desktop wallpaper, and new background color.

Although the date and time is remembered, my time zone is reset back to the default New York on every boot. This causes some confusion with email date/time stamps. Is there a cure for this? I would much prefer to remain permanently in Johannesburg.

Thanks for the great software.
ted



First, a more general answer: The system time in a Unix system is composed of elements from two resources: The interpretation of the battery-backed clock (which exists on most Intel/AMD architecture boards) and the localization settings, which determine the time offset to Greenwich mean time (GMT), or rather “Universal time” (UTC). Although your question is more about localization, I’ll look at the battery-backed clock setting first.

Under Unix/Linux, it is common always to leave the BIOS/real-time clock time in Universal time (UTC) and let the system time be set by time zone settings automatically during boot. Under Windows it seems common instead to have the real-time clock in “local time,” and even rewrite the real-time clock time during the daylight savings time switch. It is possible for both operating systems to change the default “BIOS” time interpretation to either

“localtime” or UTC; however, it’s probably easier to do this under Linux than it is to search for an appropriate setting or registry patch or change a system setting in Windows.

When a Linux system reads the time from the battery-backed real-time clock, it honors the file `/etc/adjtime`, which contains the word

```
UTC
```

in its last line if the BIOS time is expected in Universal time, or

```
LOCAL
```

if the BIOS time is stored as local time instead.

Changing this setting in `/etc/adjtime` will change Linux behavior when reading the battery-backed clock with `hwclock -s` during system start.

GNU/Linux systems will not write back their own system times to the BIOS automatically, unless instructed to do so during system shutdown, whereas Windows will do this frequently whenever time needs to be adjusted because of a winter or summer time change, so you might still experience differences when dual booting and it’s time to switch between daylight savings and standard time.

Now the offset between this time reference and the local time adjustment determined by the time zone is set by a `timezone` file stored as `/etc/timezone`. Normally, this file is created just once during system installation when time zone settings are interactively queried,

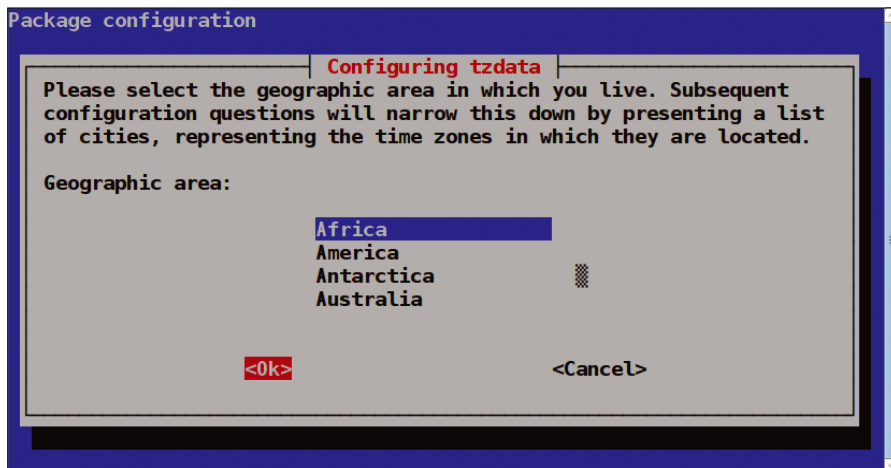


Figure 1: Choosing the geographic area for tzdata.

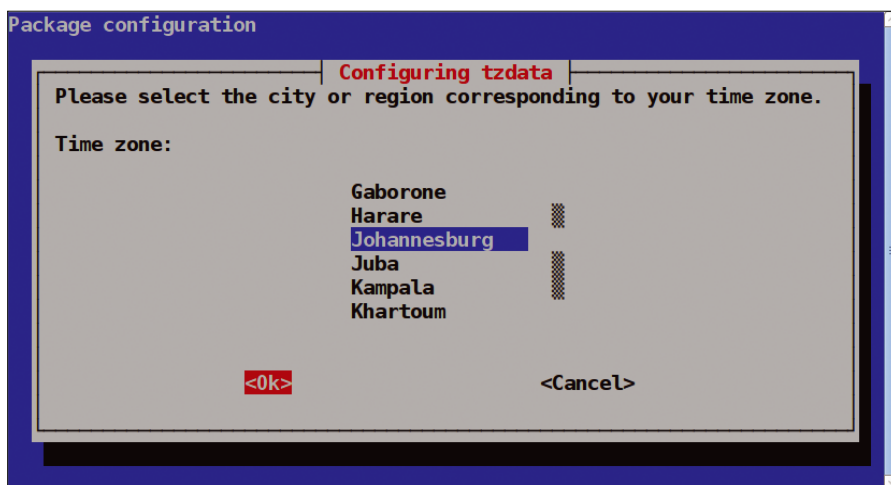


Figure 2: Choosing the region for tzdata.

```
$ sudo dpkg-reconfigure tzdata
Current default time zone: 'Africa/Johannesburg'
Local time is now:      Wed Oct 5 03:06:27 SAST 2016.
Universal Time is now: Wed Oct 5 01:06:27 UTC 2016.

$ ls -l /etc/timezone
-rw-r--r-- 1 root root 20 Okt  5 03:06 /etc/timezone

$ cat /etc/timezone
Africa/Johannesburg
```

Figure 3: Setting and checking the timezone configuration from the shell.

which can also be done at a later point (Figures 1-3) using the Debian command

```
sudo dpkg-reconfigure tzdata
```

Because Knoppix is distributed as a Live system, the standard setting is to interpret the battery-backed clock time in local time “as is,” with no offset to universal time. For easy changeability for non-Unix experts (who most likely don’t know about the settings mentioned above), the timezone setting offset is guessed from the lang variable (lang=en, lang=de, etc.) or set as an

additional tz boot command-line option, such as

```
knoppix64 tz=Africa/Johannesburg
```

either typed on the boot screen for the current session or written to the APPEND option lines in boot/syslinux/syslinux.cfg (boot/syslinux/syslinux64.cfg in the UEFI boot variant for 64-bit and boot/syslinux/syslinux32.cfg for the UEFI 32-bit variant) on the Knoppix USB flash disk:

```
APPEND lang=de apm=power-off ⌘
initrd=minirt.gz nomce libata.force=⌘
noncq hpsa.hpsa_allow_any=1 ⌘
loglevel=1 tz=Africa/Johannesburg
```

This is also (partly at least) documented in the knoppix-cheatcodes.txt file found in the KNOPPIX directory. To be on the safe side, you might want to add the timezone setting tz=area/region to all APPEND lines in the .cfg files.

For an uncompressed Knoppix hard disk installation, the same setting can be placed in the KERNEL configuration statement in the GRUB configuration file /boot/grub/menu.lst.

Now your special setup of the boot-only CD with a USB flash disk is only a little more complicated, because the boot settings are read from the boot-only CD’s isolinux settings in boot/isolinux/isolinux.cfg, which is not easily changeable without remastering the CD.

For creating a new boot-only CD with changed settings, please do the following (Listing 1).

1. Mount the CD.
2. Copy the CD content to a directory.
3. Make the copied content writable.
4. Change settings, and then save the file (Ctrl + X in Nano).
5. Create a new bootable ISO (if your system does not have the mkisofs command, use genisoimage instead).
6. Burn the new ISO image to a CD medium (use wodim if cdrecord is not available).

The new boot-only CD now contains the settings for your time zone. ■■■

LISTING 1: Creating a CD with Changed Boot Options

```
01 mount /media/sr0
02 cp -a /media/sr0 boot-cd
03 chmod -R +w boot-cd
04 nano boot-cd/boot/isolinux/isolinux.cfg
05 mkisofs -pad -l -r -J -no-emul-boot -boot-load-size 4 -boot-info-table
   -b boot/isolinux/isolinux.bin -c boot/isolinux/boot.cat -hide-rr-moved
   -o boot-cd.iso boot-cd
06 cdrecord -v boot-cd.iso
```



Developing apps with Apache Cordova

App Factory

Roll out an app elegantly and quickly for up to eight operating systems using the Cordova framework. According to the Apache Foundation, the only requirements are knowledge of HTML, CSS, and JavaScript. *By Tim Schürmann*

In early 2009, the small company Nitobi introduced a software tool named PhoneGap. Web developers could use this tool to develop complete apps in HTML, CSS, and JavaScript that simultaneously ran on iPhones, Androids, and BlackBerrys. Only two-and-a-half years later, Adobe took over Nitobi, and PhoneGap along with it. Adobe passed the source code on to the Apache Foundation, which is developing the framework under the name Cordova [1] (see the “Oh, You Beautiful Cordovas” box).

OH, YOU BEAUTIFUL CORDOVAS

Cordova is a small town in an idyllic location on the Gulf of Alaska. Geographers will find another place of the same name in central Alabama. Cordova is also the name of a suburb of Memphis, Tennessee. However, it is much more likely that you will know the Spanish city of Córdoba in Andalusia, which is also capital of the Province of Córdoba. In English, Córdoba is usually written as Cordova.

The takeover did not change the basic way the framework functions. As a developer, you can program a complete app in HTML, CSS, and JavaScript in Cordova, which then runs on iOS, Android, BlackBerry 10, Windows, Firefox OS, WebOS, Ubuntu Touch, and even as a normal web application in the browser. Cordova also supports Amazon’s FireOS, although this option is considered deprecated.

Practical Testing

The source code is subject to Apache License 2.0. Cordova has already helped launch numerous apps, including fitness app Sworkit [2] and the baby monitoring software Clever Baby [3]. Many other developer tools harness the Cordova framework and development environment, such as Ionic [4] and PhoneGap by Adobe [5]. The company continues to use the original name of the project.

Toolkit

First, Cordova requires the desired target platforms’ SDKs for its work. If you want

to offer your app on Android devices, you have to install the Android Studio or Android SDK (aka Android command-line tools) in addition to a Java Development Kit (JDK) [6]. In the case of Android, some environment variables should also be set; for example, `JAVA_HOME` prompts the JDK installation, `ANDROID_HOME` is the directory with the Android SDK, and `PATH` should contain the `tools` and `platform-tools` sub-directories from the Android SDK.

Node in the Bag

A range of command-line tools make it easier to build apps as a Cordova developer. These tools are based in turn on Node.js [7], which is found in most distros’ repositories. Cordova expects the framework under the program name `node`, although Ubuntu calls the program `nodejs`. To remedy this, additionally install the `nodejs-legacy` package, which adds a matching symlink to `nodejs`.

The `npm` command, which is the Node.js package manager installs Cordova with a simple:

```

tim@ubuntu: ~/cordova/LinuxMagazin
tim@ubuntu:~/cordova/LinuxMagazin$ cordova requirements

Requirements check results for android:
Java JDK: installed 1.8.0
Android SDK: installed
Android target: not installed
Please install Android target: "android-23".

Hint: Open the SDK manager by running: /home/tim/android-sdk-linux/tools/android
You will require:
1. "SDK Platform" for android-23
2. "Android SDK Platform-tools (latest)"
3. "Android SDK Build-tools (latest)"
Gradle: installed

Requirements check results for browser:
Check failed for browser due to Failed to check requirements for browser platform.
m.check_reqs module is missing for platform. Skipping it...
Error: Some of requirements check failed
tim@ubuntu:~/cordova/LinuxMagazin$

```

Figure 1: Cordova recognizes that the developer has the Android SDK installed; however, the libraries for Android 6.0 (API level 23) are missing.

```
sudo npm install -g cordova
```

A short moment later, you can create a first small sample application.

Hello Worlds!

To create a new project, use the Cordova tools in a working directory, which I here dubbed LinuxMagazin:

```
cordova create LinuxMagazin
```

When first called, Cordova asks whether it can transmit telemetry data to its developers. You say no in this dialog by pressing *n*. For the next step, switch to the project directory created by Cordova and add the platform on which the app will run. In this example, I began with the browser:

```
cd LinuxMagazin
cordova platform add browser
```

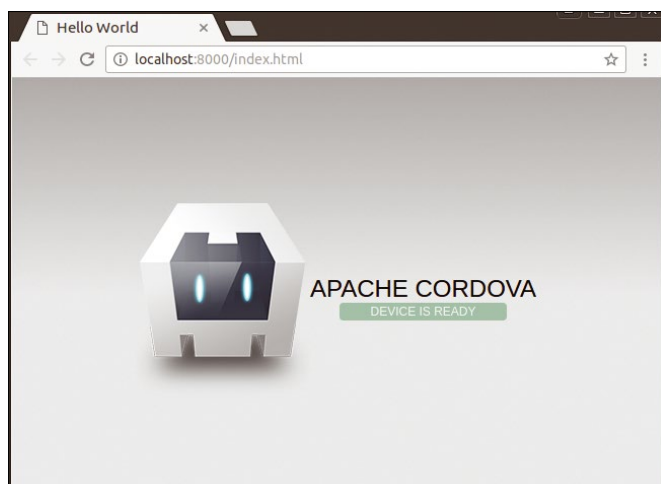


Figure 2: The Cordova app also runs on request as a web application in a web browser.

You can add more platforms with the same command, such as Android (`cordova platform add android`). The `cordova platform` command spits out a list of all possible platform names. The configuration file `config.xml` collects all the project's basic settings. It is located in the project directory and, among other things, contains the name and a (short) description of the app.

If the project has been set up, the developer should still use `cordova requirements` to check whether the computer really fulfills all the necessary SDKs and software requirements (Figure 1).

Old Friends

The `cordova create` command automatically creates a simple sample app, which can be seen in the image

for Figure 2. Like any Cordova app, this consists of a normal website that bears the `index.html` name by default and is found in the `www` subdirectory of the project directory – in this example, under `LinuxMagazin/www/index.html`. You can adapt and expand this at will. The `cordova create` command also creates the `css` directories for all style sheets, as well as `img` for pictures and `js` for JavaScript code. You can change this directory structure as you like, as long as all the app content remains in the `www` directory.

An HTML rendering engine supplied by Cordova (Figure 3) displays finished websites on the smartphone. WebView offers several special JavaScript functions that allow developers to access the smartphone hardware, as well as other items (see the “Functional and Productive” box).

The WebView provides only a few core functions, which plugins can then expand (Figure 4). A plugin makes it possible, for instance, to query contacts, whereas another allows access to the camera. Thanks to the plugin option, you can flexibly add further functions and address new hardware, but the app only carries programming code for the functions it really requires. The Cordova project itself is developing some plugins for the most important functions, including access to the camera, contacts, or the geolocation API.

FUNCTIONAL AND PRODUCTIVE

The smartphone is accessed on the app via the relevant JavaScript functions. For example, the following command will make the phone vibrate for one second:

```
navigator.vibrate(1000);
```

Cordova provides this JavaScript function.

Additionally, the framework triggers a JavaScript event in certain situations. For example, the `menubutton` event occurs if you press the menu button on your smartphone, which you intercept with a suitable event listener:

```
document.addEventListener("menubutton", onMenuKeyDown,
false);function onMenuKeyDown() {
// Button pressed!
}
```

In this example, the code calls `onMenuKeyDown()` as soon as the smartphone user presses the menu button. Some events, however, occur only on specific operating systems. The `menubutton` event is absent from iOS and Windows Phone, for instance. All the Cordova events and functions are listed in the Cordova documentation [8].

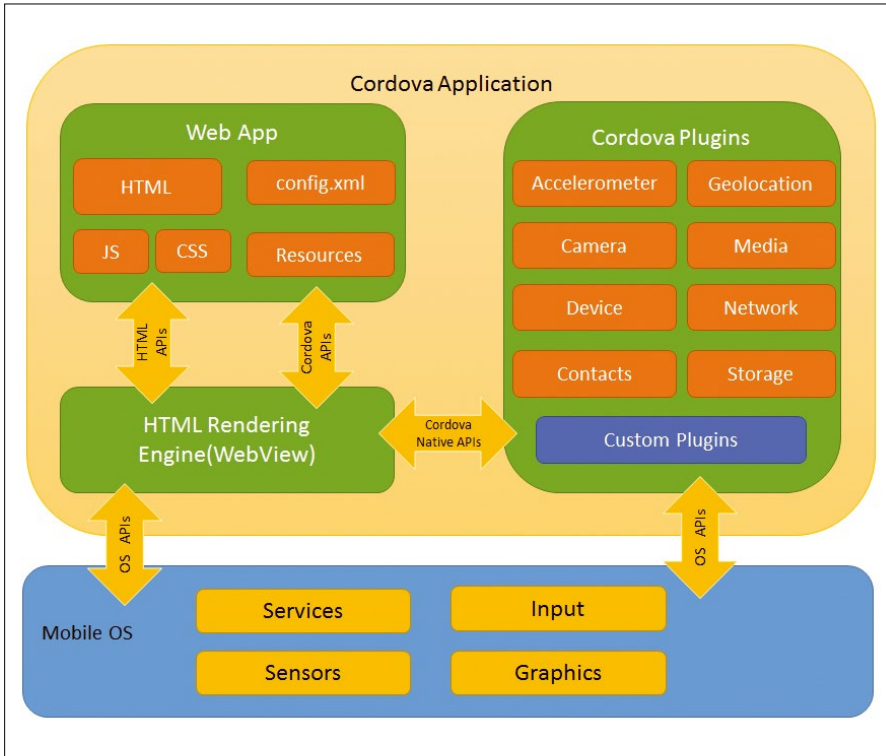


Figure 3: The web application runs a WebView, which can be extended with plugins. In turn, the plugins access the hardware or operating system functions directly.

In a newly created Cordova project, the programmer's first step must be to activate the required plugins individually:

```
cordova plugin add cordova-plugin-vibration
```

In this example, Cordova added the plugin for a vibrating alarm. The Apache Foundation has a directory of available plugins on its site [9], all of which can be added with the `cordova plugin add` command. Technically speaking, it runs `npm` in the background on the hard drive.

If you attach the `--save` parameter to the command, Cordova enters information, including the version number of the plugin, in the `config.xml` file. To return to a known state with precisely the plugin versions originally used, you enter the `cordova prepare` command.

Test Run

The `cordova run <browser>` command starts the finished app. In place of `<browser>`, you use the name of the platform on which the app is intended to run. If `browser` is used, Cordova automatically boots a minimalistic web server in the background that awaits requests by default on `localhost` port

`8000`. You then access your app by typing `http://localhost:8000/index.html` into the URL bar of the browser.

When the web server runs, Cordova starts the Chrome browser (Figure 2),

unless it is not installed, in which case Cordova does not run the app, and the web server is promptly stopped. As a result, your only option as an app developer is to install Chrome or do without the browser as a possible platform. If you close Chrome, you must then force the web server to shut down by pressing `Ctrl + C`.

If the browser is not the target, Cordova starts the app on the appropriate platform, which runs on either a plugged-in device or an emulator. Running

```
cordova run --list
```

lists all the existing devices and emulators. You can explicitly choose one of the devices with `--target`; in the following case, this would be `MyAndroidVM`:

```
cordova run android --target=MyAndroidVM
```

To start it on an emulator, a shortcut boots a virtual Android platform,

```
cordova emulate android
```

This command also works for other platforms, with a `--list` displaying their respective version numbers in detail.

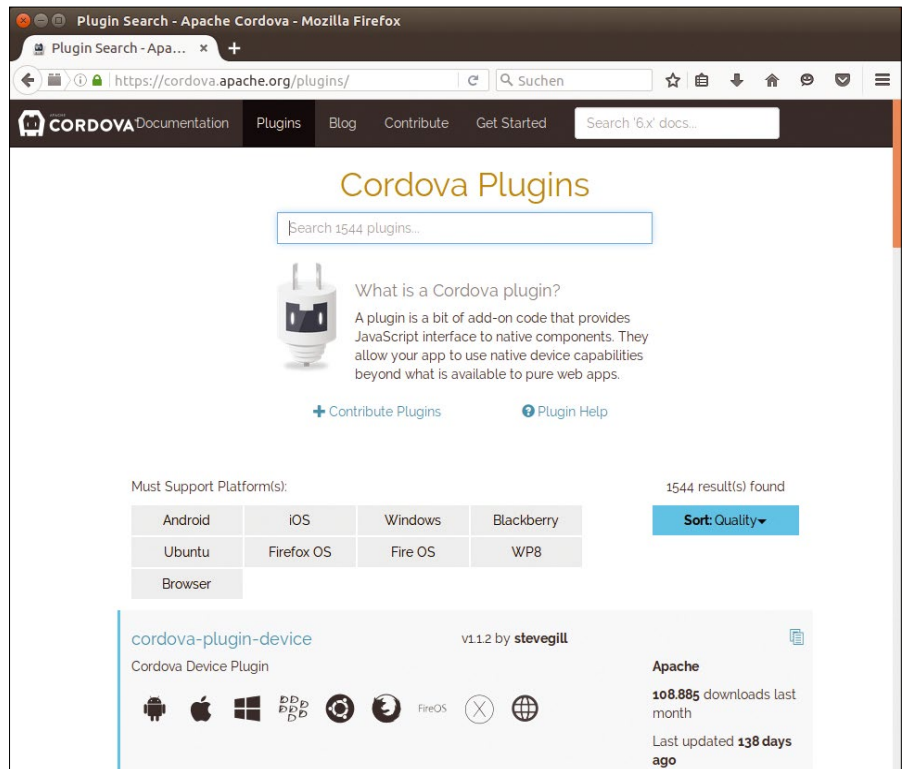


Figure 4: Some plugins can handle in-app sales. Before installing a plugin of this sort, however, you should find out the platform on which it runs.

Scarcity

The Cordova support ends here, however. In particular, the framework was not designed to offer any user-interface components or widgets that would allow you to click together a user interface quickly. Only simple dialog windows for warnings, error messages, and simple queries are available (via the `cordova-plugin-dialogs` plugin).

For further UI elements (e.g., switching pages), you must either implement them yourself or rely on JavaScript libraries such as the popular jQuery UI [10]. However, a disadvantageous side effect is that the application will not be able to assume the appearance of the target platform.

Another disadvantage for pure Linux users: Because Cordova requires the SDKs for each respective target platform, iOS apps can only be created and tested on OS X. Therefore, if you want to develop for iPhones, you need an Apple computer. The same applies to Windows Phone and a Windows computer.

Furthermore, app developers must set up and maintain the SDKs by hand, as well as manually certifying apps such as those for the Play Store. The Cordova documentation consists only of a short tutorial, though it is still a very detailed reference [1].

Conclusion

Despite these restrictions, HTML, CSS, and JavaScript experts can get to their first app fairly quickly using Cordova. You can address the smartphone hardware in the familiar JavaScript and do not need to worry about platform differ-

ences: The same code runs on both iOS and Android.

Cordova's command-line tools abstract the individual platform SDKs from the properties, although app developers do require the relevant hardware for their installation. You must also work on the command line, implement the user interface completely independently, and retrofit numerous functions via the appropriate plugins.

If you want to create a finished app at greater speed or with just a few mouse clicks, then, you will be better served by another tool. ■■■

INFO

- [1] Cordova: <https://cordova.apache.org>
- [2] Sworkit: <http://sworkit.com>
- [3] Clever Baby: <http://www.myclivebaby.com>
- [4] Ionic: <http://ionicframework.com>
- [5] Adobe PhoneGap: <http://phonegap.com>
- [6] Android Studio: <https://developer.android.com/develop/index.html>
- [7] Node.js: <https://nodejs.org>
- [8] Cordova documentation: <https://cordova.apache.org/docs/en/latest/>
- [9] Apache's plugin directory: <https://cordova.apache.org/plugins/>
- [10] jQuery UI: <https://jqueryui.com>

What?! Archives come with my digital subscription?

That's a lot of articles!



Archives + Current!

Sign up for a digital subscription to get the latest issues of Linux Magazine, PLUS access to archive articles.

shop.linuxnewmedia.com/digisub



cURLing up to file transfers

Gone Curling

cURL is a powerful file transfer tool that is simple to learn but can get you in trouble if you don't keep track of your options. *By Bruce Byfield*

First released in 1997, cURL (pronounced “see URL” or “curl”) [1] is a relatively new tool for the command line. However, it quickly became a standard command-line tool as the need to transfer data between servers or from an Internet site grew. Moreover, copying a directory has become a standard way to install scripts written in interpretive programming languages like Python. Today, cURL is now included in the default installation of most Linux distributions. Functionally, it resembles `wget` [2], although the two commands differ in syntax.

The basic structure of `curl` is simple, but can quickly become complex as you add options for the many transfer protocols it supports. Even the command's man page notes that “the number of features will make your head spin!” And, with cURL's proxy support, user authentication, FTP upload, HTTP POST, SSL connections, cookies, and file transfer resume, the complexity of its options can quickly become apparent.

Fortunately, however, many of cURL's options are designed for using specific protocols in specific circumstances and are often unnecessary. Moreover, some of the supported transfer protocols, like

GOPHER and TELNET, are frankly obsolete, and many users are likely to use, if not the default HTTP, then HTTPS or SSH, with forays into FTP and one or two others.

Downloading and Uploading

Like any command that involves copying, cURL requires a data source and a target – standard output, if no target is specified, or a specifically named file. As the name of the command implies, the data source is a URL – a protocol that starts with the name of the protocol, followed by an address.

```
bb@nanday:~$ curl designingwithlibreoffice.com
<!DOCTYPE html>
<html lang="en-CA">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width" />
<title></title>
<link rel="profile" href="http://gmpg.org/xfn/11" />
<link rel="pingback" href="http://designingwithlibreoffice.com/xmlrpc.php" />
<!--[if lt IE 9]>
<script src="http://designingwithlibreoffice.com/wp-content/themes/misty-lake-wpcom/js/html5.js" type="text/javascript"></script>
</endif-->
```

Figure 1: On the Internet, you can usually enter an abbreviated command and leave cURL to figure out the transfer protocol.

If you do not specify a protocol, cURL will try to guess a suitable one. Consequently, when you use a public Internet site, which almost certainly uses HTTP or HTTPS, you can probably enter an abbreviated form of the URL (Figure 1); for example:

```
curl designingwithlibreoffice.com
```

However, in some cases, you might need to specify a transfer protocol. That is especially likely if you want to specify the secure HTTPS protocol version of a site by entering one or the other of these commands:

```
curl https://designingwithlibreoffice.com
curl --request GET https://designingwithlibreoffice.com'
```

Notice the quotation marks in the structure for the `--request` option.

Both these commands print the web page out below the command in the terminal, which is convenient when you want to check information without downloading, although you might want to pipe the command through `less` for easier viewing. To save the download by specifying a target file,

```
curl -o /home/bb/home.html http://designingwithlibreoffice.com
```

would save the web page to the file `home.html`. If you wanted to add a directory structure, then you specify `--create-dirs`, and `curl` will create any directories that do not already exist in the output file's path. You can even

use `--remote-name (-O)` to save to a third server or site.

Uploading uses a similar structure, except that `--upload-file FILE (-T FILE)` replaces `-o FILE` and `--request PUT` replaces `--request GET`. You could also use `--data DATA (-d)` with `--request POST` when uploading a plain text file, or `--data-binary` for a binary upload.

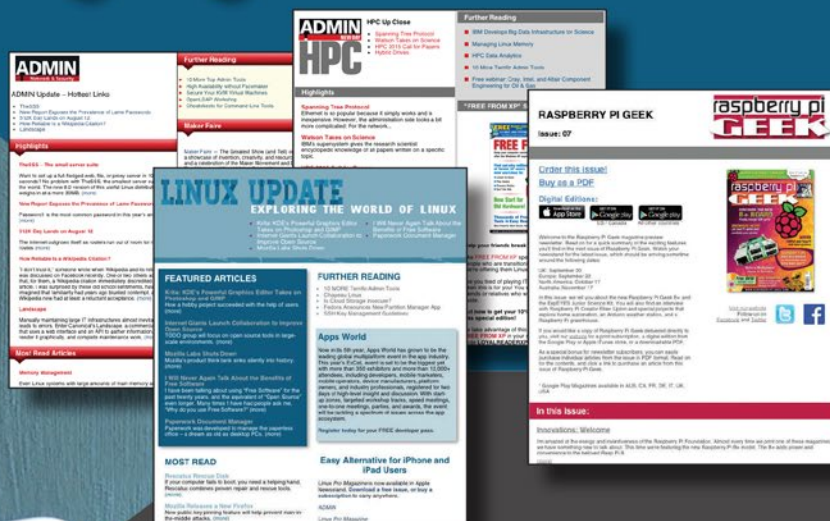
In both downloading and uploading, cURL displays a progress meter until its operation is complete. Alternately, you can use `-#` to display a progress bar consisting of hash tags (Figure 2). To use neither, add the `-s` option, but beware that the option will also mute any other messages from cURL.

cURL can move multiple files using space-separated lists. However, to save to file, you must repeat the `-o` or `-T` op-

```
bb@nanday:~/Downloads$ curl -o /home/bb/home.html http://designingwithlibreoffice.com
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left   Speed
100 24765  100 24765    0     0  27255    0 --:--:-- --:--:-- --:--:-- 27244
bb@nanday:~/Downloads$ curl -# -o /home/bb/home.html http://designingwithlibreoffice.com
##### 100.0%
bb@nanday:~/Downloads$
```

Figure 2: cURL displays either a detailed progress meter or a simpler bar while operating.

IT Highlights at a Glance



Too busy to wade through press releases and chatty tech news sites? Let us deliver the most relevant news, technical articles, and tool tips – straight to your inbox.

Linux Update • ADMIN Update • ADMIN HPC • Raspberry Pi
Keep your finger on the pulse of the IT industry.

Admin and HPC: www.admin-magazine.com/newsletter

Linux Update: www.linuxpromagazine.com/mc/subscribe

Raspberry Pi: www.raspberry-pi-geek.com/mc/subscribe

tion, specifying a unique file name for each target. Otherwise, each file in turn will overwrite the target file specified in the first element of the list. For example:

```
curl -o /home/bb/home.html ⚡
http://designingwithlibreoffice.com/⚡
home.html -o /home/bb/contact.html ⚡
http://designingwithlibreoffice.com/⚡
contact.html
```

cURL also supports a limited version of regular expressions. For example,

```
curl ⚡
http://designingwithlibreoffice.com/⚡
{home, contact, review}.html
```

Alternatively, if targets are regularly named, you can use alphanumeric sequences of numerals or upper- or lowercase letters enclosed in square brackets:

```
curl ⚡
http://designingwithlibreoffice.com/⚡
images[1-100].png
```

```
curl ⚡
http://designingwithlibreoffice.com/⚡
images[a-z].png
```

If a source or target is password protected, cURL will stop operating unless you supply authentication. Simply `--user USER (-u)` with HTTP will make cURL prompt for a password. With other protocols, you should use the format `--user USER:PASSWORD`, but only if the username does not include a colon (:). In SSL/SSH, the option `--pass [PHRASE]` can be used to enter a passphrase. Similarly, if you need to log in for a proxy, add the option `--proxy-user USER:PASSWORD (-U)`.

Other options control how cURL carries out a command. For instance, if you choose, you can specify the maximum file size to use and maximum time to run with `--max-file-size [BYTES]` and `--max-time [SECONDS]`. Should an operation time out, you can set the time cURL waits for it to resume with `--connect-timeout [SECONDS]`. If an operation has failed entirely, you can use `--continue-at [BYTES] (-C)` to instruct cURL to try to resume where it left off, although you should check to see if a file or two was missed around the place where the operation timed out.

When using FTP, you also have the `--quote COMMAND (-Q)` option, with which you can send a command once you have made a connection. Among the dozen or so commands that you send are `chown`, `chgrp`, `chmod`, `mkdir`, `password`, `rm`, and `dir`, all of which may help you manipulate files and continue an operation.

Transfer Protocol Options

Many of cURL's options define the version of a protocol to use. These options are mostly useful if normal use of cURL fails, such as when cURL fails to identify the protocol. Such failures are most likely when a command has multiple entries and you need to specify a change from one protocol to another.

On the Internet, you may never need most of the available options, because cURL defaults to HTTP, the most widely popular transfer protocol. The sole exception may be `--location (-L)`, which directs HTTP- or HTTPS-based operations to follow a redirection when a URL has moved. cURL also has a `--location-trusted` op-

tion, which will send the name and password to a new URL, but it should not be used unless you know that it can be trusted.

SSH/SSL also has one or two options, the most useful of which is probably `--pubkey KEY`, which gives the path to your public encryption key. At times, the `--ssl` option is useful, allowing a command to continue with a less secure protocol when SSH is not supported.

However, the transfer protocol to watch most closely is FTP – and not just because it is aging and not as secure as SSH or HTTPS. If you are using FTP, my advice is to refer to the man page constantly. Often, FTP will be an exception to an option or behave somewhat differently than other protocols. If you must use FTP, then include the option `--ftp-pasv`, which runs the protocol in its securest mode – but remember that security is relative.

A Final Caution

You can learn cURL's basics in 20 minutes. However, as the command structure balloons, take extra time to check the syntax, especially when multiple files or operations are included in a single command.

The trouble is that cURL takes the last option used. This format, while logical, can lead to unexpected results. For example, when multiple files are listed, if you do not specify an option before each file name, a file will use whatever options the file listed before it used. In particular, this habit can easily cause a file to overwrite the files saved before it. Similarly, if you are not careful, the same format can result in cURL suddenly using a transfer protocol or behavior that you did not intend, possibly creating security and functional problems.

The best solutions are either to keep your uses of cURL simple or be as specific as possible, giving paths in full and checking the syntax before pressing the Enter key. Without one of these tactics, cURL can easily seize control and switch from a useful servant to a capricious master. ■■■

INFO

[1] cURL: <https://linux.die.net/man/1/curl>

[2] wget: <https://linux.die.net/man/1/wget>

**LOST YOUR BOOKSTORE?
LET US BE YOUR BOOKSTORE**

Browse our shop for single issues of *ADMIN*, *Linux Pro*, *Linux Magazine*, *Raspberry Pi Geek*, *Drupal Watchdog* and *Ubuntu User* – delivered right to your door.

- shop.linuxnewmedia.com/single

Better yet, subscribe, and you won't need a bookstore.

- shop.linuxnewmedia.com/subs

shop.linuxnewmedia.com
DIGITAL AND PRINT EDITIONS AVAILABLE!

Shop the Shop

shop.linuxnewmedia.com

Want to subscribe?

Searching for that back issue you really wish you'd picked up at the newsstand?

Discover the past and invest in a new year of IT solutions at Linux New Media's online store.

shop.linuxnewmedia.com

DIGITAL & PRINT SUBSCRIPTIONS



SPECIAL EDITIONS





Termux for Android

Tiny Linux

Termux transforms an Android device into a lightweight and versatile Linux environment. This article introduces you to this must-have app. By Dmitri Popov

Running a regular Linux distribution on an Android device is not a new idea, and several apps in the Google Play Store will let you deploy a full-blown distribution on an Android smartphone or tablet. If your device is powerful enough and storage space is not an issue, there's nothing wrong with this approach. In most cases, however, having a full-size Linux distribution crammed into an Android machine is overkill.

Enter Termux [1], an app that provides an alternative way to use popular Linux tools on an Android device. Combining a terminal emulator and a lightweight Linux environment in one tiny 157K app (Figure 1), Termux puts a wide range of Linux tools at your fingertips: from Bash, Emacs, and Git to curl, rsync, and OpenSSH. The app requires no root privileges, and the dedicated Termux:API add-on gives you access to Android's APIs, so you control your Android device from the command line and shell scripts.

First Steps

Termux is available on Google Play Store and F-Droid [2], so you can install it from your preferred source. The app's default functionality can be extended by installing add-ons. The Termux:API

add-on, for example, adds support for Android API, whereas the Termux:Styling package lets you pick an alternative color scheme and font. The Termux:Widget add-on gives you a home screen widget containing short-

```

Preparing to unpack ../archives/glib_2.48.2-2_arm.deb ...
Unpacking glib (2.48.2-2) over (2.48.2) ...
Preparing to unpack ../ncurses_6.0.20160423-1_arm.deb ...
Unpacking ncurses (6.0.20160423-1) over (6.0.20160423) ...
Preparing to unpack ../termux-tools_0.29_all.deb ...
Unpacking termux-tools (0.29) over (0.28) ...
Setting up ca-certificates (20160907) ...
Setting up glib (2.48.2-2) ...
Setting up ncurses (6.0.20160423-1) ...
Setting up termux-tools (0.29) ...
$ apt-cache show dcraw
Package: dcraw
Architecture: arm
Installed-Size: 272
Maintainer: Fredrik Fornwall <fredrik@fornwall.net>
Version: 9.27.0
Description: Raw digital camera images decoding utility
Description-md5: 52ade4761300c51ea577548bceb96cdb
Homepage: http://www.cybercom.net/~dcoffin/dcraw/
Depends: littlecms, libjasper
Filename: dists/stable/main/binary-arm/dcraw_9.27.0_arm.deb
Size: 105916
SHA256: 516318b9c84a946471410bfc7279b964759b6ef2612025b0ee17a2fcbd6f7784
$

```

Figure 1: Termux gives you a lightweight Linux environment on Android.

cuts to shell scripts, so you can run scripts in Termux with a single tap. You can install these add-ons for free from F-Droid. If you choose to install the add-ons from Google Play Store, you have to purchase them. Think of it as a way to donate to the project and support its developer.

Working in the terminal using the on-screen keyboard can quickly become tedious. Fortunately, Termux provides shortcuts to many common keyboard actions. Most of these shortcuts are identical to those in the Bash shell. For example, the Ctrl + A and Ctrl + E shortcuts move the cursor to the beginning and the end of the line, respectively, and the Ctrl + C and Ctrl + D shortcuts abort the current process and log out of the terminal session.

The only difference is that Termux uses the Volume Down hardware button instead of the Ctrl key. The Volume Up button in combination with various letters is used to emulate keyboard keys. The Volume Up + T shortcut emulates the Tab key, whereas Volume Up + W, Volume Up + A, Volume Up + S, and Volume Up + D act as arrow keys. The *Using a touch keyboard* page [3] provides a complete list of all supported shortcuts, and it's worth perusing before you start working in Termux (Figure 2).

There are three things you need to do when running Termux for the first time: Update the app's software repository, install the desired packages, and configure storage access. Termux uses the apt tool for package management, so if you are coming from Debian or Ubuntu, you should feel at home in Termux.

To update the repository, run the apt update command. Want to see all packages in the repository? Run the apt list command. To find a specific package, use the apt search PACKAGE command, and to install the package, issue the apt install PACKAGE command. Upgrading the installed packages is a matter of running the apt upgrade command, and the apt remove PACKAGE command can be used to remove an installed package.

What packages to install depends entirely on what you want to use Termux for, but at the very least you might want to install the nano text editor, the rsync file-copying tool, and the OpenSSH con-

nection tool. These three tools allow you to edit text and configuration files and perform backups as well as access and control remote machines. To install these packages in one go, use the following command:

```
apt install nano rsync openssh
git curl wget
```

To configure storage access, run the termux-setup-storage command. Doing this grants Termux permission to access shared storage on Android 6.0 and higher as well as create the dedicated storage folder containing symbolic links to various storage locations. This includes ~/storage/shared (the root folder shared by all apps), ~/storage/dcim (the standard folder for photos and videos), and ~/storage/downloads (the default folder for downloaded files).

Practical Uses

Having a tool-ed-up Linux environment on your Android device lets you perform tasks that usually require third-party commercial apps. For example, instead of using a specialized app, you can connect and control a remote Linux machine by running the ssh user@host command (provided an SSH server is also installed on the remote host). And, using rsync, you can easily set up a robust solution for backing up data on your Android device to a remote machine.

To do this, it's a good idea to enable password-less SSH login, so you don't have to type the password every time you perform a backup. Start with run-

ning the ssh-keygen command. Skip password creation when prompted by pressing Enter. Once a key pair has been created, run the ssh-copy-id -i .ssh/id_rsa.pub user@host command (replace user and host with the username and the remote machine's IP address or domain name). This copies the public key to the remote machine, thus enabling passwordless login. Backing up data on your Android device to the remote server is a matter of using an appropriate rsync command. If you want to back up photos and videos, the following command will do the trick:

```
rsync -avz -e ssh ~/storage/
dcim user@host:/path/to/backup/dir
```

Instead of running this command manually every time you need to perform a backup, you can write a simple Bash shell script, save it in the ~/.shortcuts directory, and add a Termux widget that links to the script. This way, you can trigger a backup operation directly from the home screen.

Staying on the subject of photos, the jhead utility makes it possible to process photos stored on your Android device. Run the apt install jhead command to install the utility in Termux. Using jhead, you can replace generic filenames with something more meaningful, like timestamps. The following command renames all .jpg photos using the yearmonthdate-hourminutesecond rule (e.g., DC_0001.jpg becomes 20161010-152557.jpg):

```
for file in *.jpg; do jhead
-n%Y%m%d-%H%M%S $file; done
```

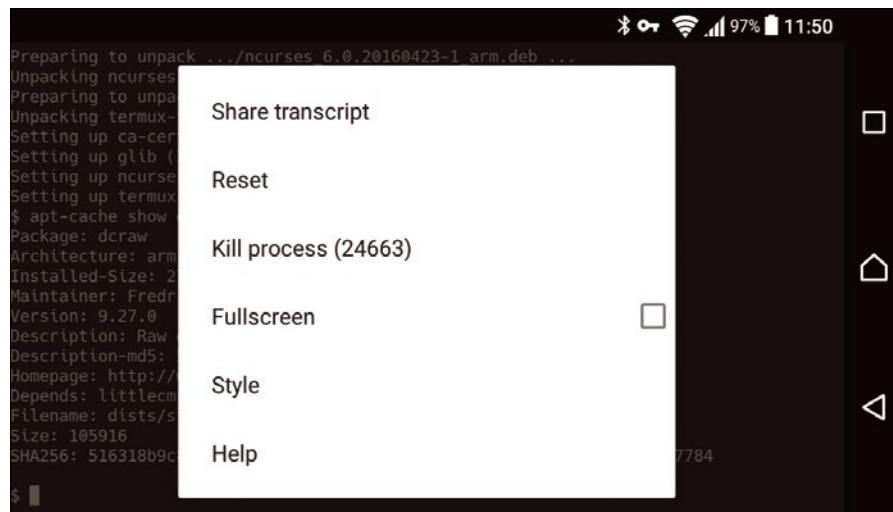


Figure 2: Termux's online help is only a tap away.

If you happen to shoot in RAW, you can convert RAW files into JPEG photos on your Android device using a combination of DCRaw, Netpbm, and ImageMagick. Install these tools by running `apt install dcrw netpbm imagemagick`. Then, use the following command to convert a single RAW file to the PNG format:

```
dcrw -c foo.RAW | pnmtopng > foo.png
```

In this command, the `dcrw` tool decodes the RAW file and pipes the output to the `pnmtopng` tool, which saves it in the PNG format. You can then use the `convert` tool to convert the resulting PNG file into JPEG:

```
convert foo.png foo.jpg
```

To process all RAW files and save them in the JPEG format, use the following commands:

```
for file in *.RAW ; do dcrw ⌘
-c -g 2.4 12.92 "$file" | pnmtopng ⌘
> "${file%.*}.png" ; done
for file in *.png ; do convert ⌘
"$file" "${file%.*}.jpg" ; done
```

Instead of `convert`, you can use the `mogrify` tool as follows:

```
mogrify -format jpg *.png
```

Keep in mind, though, that `mogrify` overwrites originals, so use it with care.

Want to apply effects to photos without resorting to a dedicated app? Termux and ImageMagick can handle that, too. The latter supports the Hald CLUT technology, which makes it possible to use special color lookup tables to perform color transformations.

To make use of Hald CLUT, you don't need to understand all its intricacies, but you need to do some preparatory work. Install ImageMagick on your regular Linux machine and run the `convert hald:9 hald-9.png` command to generate a Hald CLUT table. Next, open the generated `hald-9.png` file in a photo editing application, such as `digiKam`, apply the desired color corrections (curves, levels, saturation, brightness, etc.), and save the changes under a descriptive name (e.g., `hald-9-vintage.png`).

Now you have a reference Hald CLUT table that can be used as a preset. Using this technique, you can create as many presets as needed. On your Android device, move them into the folder accessible by Termux. Use the following command:

```
convert foo.JPG hald-9-vintage.png ⌘
-hald-clut foo-modified.jpeg
```

to apply the desired preset to a photo.

Using Termux:API

Although the Termux app gives you a lightweight yet functional Linux environment, the Termux:API add-on integrates it into the Android system by providing limited access to Android API. In practical terms, this means you can access and control Android hardware and system functions directly from within Termux, either directly from the command line or through shell scripts.

A simple Bash shell script in Listing 1 uses several Termux:API methods to take a photo and write an accompanying annotation to a CSV file. The script uses the `jq` tool to extract geographical coordinates, so you need to install it first using the `apt install jq` command.

The script uses the `termux-camera-photo` method to take a photo (note that this method takes a photo without any preview and feedback). The `termux-location` method obtains location data, the `jq` tool extracts latitude and longitude values, and the `termux-dialog` method displays a dialog that lets you enter a brief description or comment. All the obtained data is then saved in the `annotations.csv` text file. The `termux-toast` method is used throughout the script to display pop-up notifications.

Termux:API offers a few other useful methods. The `termux-sms-send` method lets you send SMS messages directly from Termux, and the `termux-share` method can be used to share files using Android's sharing capabilities.

In Conclusion

Nimble and flexible, Termux puts many popular Linux command-line tools at your disposal, so you can perform various actions and automate a wide range of tasks without resorting to commercial third-party apps. Add to this the ability to access Android API using the Termux:API add-on, and you have a versatile, lightweight Linux environment that can handle practically any task you throw at it. ■■■

INFO

- [1] Termux: termux.com
- [2] Termux on F-Droid: f-droid.org/repository/browse/?fdid=com.termux
- [3] Termux shortcuts: termux.com/touch-keyboard.html

LISTING 1: Snap and Annotate

```
01 #!/bin/bash
02 dir="snapshots/"
03 dt=`date +%Y%m%d-%H%M%S`
04 if [ ! -d "$dir" ]; then
05   mkdir $dir
06 fi
07 cd $dir
08 echo "Hold camera still..." | termux-toast
09 termux-camera-photo $dt.jpg
10 echo "Done!" | termux-toast
11 lat=$(termux-location | jq '.latitude')
12 lon=$(termux-location | jq '.longitude')
13 echo "Coordinates: $lat, $lon" | termux-toast
14 comment=$(termux-dialog -m -t "Comment")
15 osm="http://www.openstreetmap.org/index.html?mlat=$lat&mlon=$lon&zoom=18"
16 echo "Photo: $dt.jpg Latitude: $lat, Longitude: $lon, OSM: $osm,
    Comment: $comment" >> annotations.csv
17 echo "All done!" | termux-toast
18 termux-vibrate
```



RISE HIGHER

EACH ISSUE OF DRUPAL WATCHDOG OFFERS TOOLS, TIPS, AND BEST PRACTICES FOR BETTER DRUPAL WEBSITES.

NOW
PUBLISHED
4 TIMES
PER
YEAR!



Renew or subscribe now!

SUBSCRIPTIONS NOW AVAILABLE WORLDWIDE!

Visit <http://drupalwatchdog.com/subscribe>

The sys admin's daily grind: Yowsup

Berry Service

Sys admin Charly likes to keep up with the state of his domestic strawberry plants, so he has outsmarted WhatsApp to allow status messages to be sent through the service from a PC or Raspberry Pi. *By Charly Kühnast*

When it comes to domestic events, I use WhatsApp to keep on top: A motion detector in the bird and hedgehog house, the status of the garden irrigation system, doorbell, water level in the aquarium, and much more. The tool that sends me the message is Yowsup [1], and it's easy to install:

```
wget https://github.com/tgalal/yowsup/archive/master.zip
unzip master.zip
cd yowsup-master/
./setup.py install
```

Yowsup makes the WhatsApp API believe it is running on cellphone. You need a SIM card that is not yet registered with WhatsApp for the login process. I used a free prepaid card for that purpose. "Free" here means that there is no monthly base charge or contract but only pay-per-call. Some mobile service providers have such SIM cards on offer.

I opted for a provider, put the SIM card in the phone, and activated the card; you need to determine the MCC/MNC pairing, which you can find online [2]: The

mobile country code for Germany is 262, the mobile network code for my network provider, Vodafone, is 2. Nothing to stop me registering now:

```
yowsup-cli registration -d -E
android -m 262 -n 2 -p 491521234567
-C 49 -r sms
```

After a few seconds, you get a text message with a registration code, say, 528-142. You need this now:

```
yowsup-cli registration -d -E
android -p 491521234567
-C 49 -R 528-142
```



Figure 1: A "Hello World!" sent by Charly to the phone.

LISTING 1: yowsup-cli Registration

```
01 status: ok
02 kind: free
03 pw: VQOHbV1d0Ajd+5GKIrhVWRNzkVO=
04 price: 0,89 fl
05 price_expiration: 1470669760
06 currency: EUR
07 cost: 0.89
08 expiration: 4444444444.0
09 login: 491521234567
10 type: new
```

The output contains my login credentials in the style of Listing 1. The username is my telephone number including the country code, and the password appears in line 3. Now I can put the SIM card in a back drawer. From now on, I can send WhatsApp messages on my PC.

```
./yowsup-cli demos -l "491521234567:VQOHbV1d0Ajd+5GKIrhVWRNzkVO="
-s 491721234567 "Howdy, Post"
```

and receive them on my mobile phone (Figure 1). I have to go – my message just reached me: Strawberry harvest! ■■■

INFO

[1] Yowsup:

<https://github.com/tgalal/yowsup>

[2] List of MCCs and MNCs by country:

<http://mcclist.com/mobile-network-codes-country-codes.asp>

CHARLY KÜHNAST

Charly Kühnast manages Unix systems in the data center in the Lower Rhine region of Germany. His responsibilities include ensuring the security and availability of firewalls and the DMZ.





Ben Everard

If you're a *Linux Voice* reader, you may be wondering what this magazine you have in your hands is – I'll get to that, but first I'll start with a little background information. Three years ago, we set out to create a great Linux magazine, and to our eternal gratitude (and surprise) many of you supported us through our crowd funding campaign, and together we launched *Linux Voice*. Back then we were told we were crazy to launch a print magazine in the digital age. Perhaps we were, but for three fantastic years we defied the odds and produced a product we're proud of. I'd like to thank all of you for helping us do that.

We created *Linux Voice* because we were passionate about Free Software, and we are so grateful to you, our readers, for sustaining and supporting our vision. Earlier this year, however, a combination of factors brought our dream face to face with reality, and it became clear that *Linux Voice* would have to change to survive.

That brings us here. We still believe that the Free Software community is what makes Free Software great, and we still believe that passionate communities deserve passionate magazines. We wanted to keep going with our vision, and we decided after much consideration the best path ahead was to merge with *Linux Magazine*. By bringing together two groups of enthusiastic Linux writers and evangelists, we hope we can make the magazine the community deserves.

If you're a regular *Linux Magazine* reader, I hope you like this addition. If you want to find out more about *Linux Voice*, you'll find loads of stuff on our website (www.linuxvoice.com), including back issues of our old magazine and our podcast.

– Ben Everard



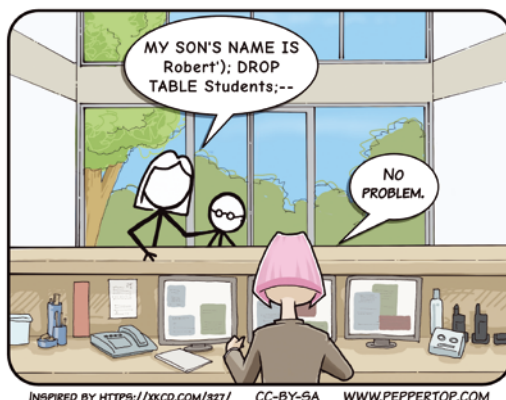
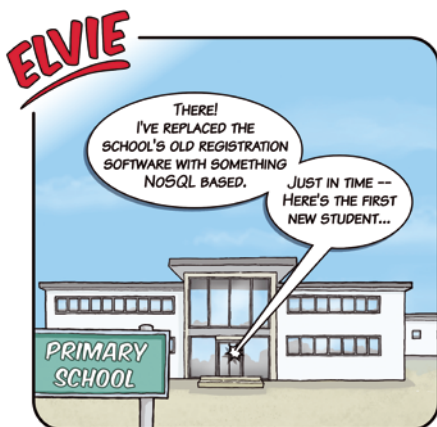
Andrew Gregory



Graham Morrison



Mike Saunders



INSPIRED BY [HTTPS://JXKCD.COM/521/](https://jxkcd.com/521/) CC-BY-SA WWW.PEPPERTOP.COM

LINUXVOICE ▶

Ownership and IoT 64
Simon Phipps
 George Orwell was wrong: Big Brother is smaller and more portable than he predicted.

Conduct Unbecoming 65
Andrew Gregory
 Why doesn't the Free Software Foundation have a code of conduct?

Editor Upgrade 66
Mike Saunders
 Mike looks inside Vim 8 – the first major release of the Vim text editor in 10 years.

NoSQL FAQ 72
Ben Everard
 Big Data is like the Matrix – Better without the sequel.

Core Tech: One ip to Rule Them All 74
Valentine Sinitsyn
 Manage and troubleshoot your network with a single command. A single, shiny, precious command.

FOSSPicks 78
Graham Morrison
 Draw cartoons with OpenToonz, build retro 3D images with VoxelShop, and explore the world with Marble 2.0 – an open source equivalent of Google Earth.

Gaming on Linux 84
Michel Loubet-Jambert
 Become a pro-sportsman from the comfort of your Linux desktop. Yes, we're serious.

Tutorial: Autodeploy Ansible Containers 86
Ben Everard
 Automate your software deployment so you've got more time to read *Linux Magazine*.

Tutorial: Multitasking with Tmux 92
Mike Saunders
 Get the benefits of window management without those pesky graphics getting in the way of your command-line bliss.

NEWS ANALYSIS

The Linux Voice view on what's going on in the world of Free Software

Opinion

Ownership and IoT

You don't control that device you think you own.



Simon Phipps is ex-president of the Open Source Initiative and a board member of the Open Rights Group and of Open Source for America.

In the Internet of Things (IoT), there's a good chance you no longer really own the things you have paid for. HP recently mounted an impressive demonstration of how and why, with lessons that reach far beyond just smart printers.

On September 13, HP was discovered to have triggered a "feature" in the firmware of their printers to make them refuse to print with ink cartridges that did not pass a check for a valid security chip. That meant that anyone whose printer firmware had previously been updated and who tried to use third-party ink cartridges for their printer found they no longer worked, even if they had before. There were a few reports of "HP original" ink cartridges triggering the lockout as well.

After a few weeks, HP announced [1] it would eventually issue an optional firmware update to undo the lockout. Notably, they didn't apologize for the lockout, nor did they promise not to repeat the exploit. They said they wanted to:

"...ensure the best consumer experience and protect [customers] from counterfeit and third-party ink cartridges that do not contain an original HP security chip and that infringe on our IP. We should have done a better job of communicating about the authentication procedure to customers, and we apologize."

In other words, next time HP decides to break your printer remotely to blackmail you into trading only with them, they will give you plenty of warning. The printer you paid for is not yours to do with as you please. It contains software which is merely licensed to you, so HP retains legal leverage over you far beyond the scope of normal hardware purchases. When you started using the printer, you agreed to surrender your rights in perpetuity. So HP doesn't feel in any way obliged to ask you; if it does, that will merely be for PR purposes.

Even that is assuming you install the updated firmware. As far as anyone can tell, HP will neither install it automatically

nor install it by default in new printers. But be thankful for small mercies. By providing a work-around to the lockout, HP is at least making sure you have a way to circumvent their restriction. That will keep you out of prison; if they had left it to you, or your supplier, to find a way out, you would very likely have been committing a criminal offense under the anti-circumvention measures in modern copyright laws.

Just doing and reversing this probably achieves their real goal, which is to scare ink suppliers. While proactive consumers will take note that HP has no regard for their freedom of choice, the majority will continue to make printer purchases based on the cost of the printer.

But it's not just printers. This is the reality of the whole world of IoT and "Smart" devices. They are all throwaway containers in which the licensed works of your suppliers reside. Back in the old days, you used to buy things in the store and own them. All the powers of the manufacturer were exhausted on purchase. But by inserting copyrighted works into their goods, manufacturers are able to require you to consent to a copyright license agreement. The terms it contains aren't just about copyright. They include liability waivers, agreements not to sue, agreement not to hack, consent to intrusion, and more. The terms of the license also

Back in the old days, you used to buy things in the store and own them.

survive way beyond the purchase of the product, into the ever-extending future of copyright durations.

When you use a device that's filled with proprietary software, you have only the rights the manufacturer chooses to bless you with. What better reason could there be to pursue devices that work with open source firmware and services? Don't just buy on price; buy on freedom, too [2]. ■■■

Info

[1] <http://www8.hp.com/us/en/hp-news/blog/Small-Business-Printing/best-possible-printing-experience.html>

[2] <http://webmink.com>

Conduct Unbecoming

Should the FSF be thinking about a code of conduct? **BY ANDREW GREGORY**

This month past there's been a bit of a spat between the lead developer of the Libreboot project and the Free Software Foundation (FSF). The cause of the disagreement is pretty unambiguously serious: The alleged bullying of a transgender member of staff by a person or persons unnamed in the employ of the Free Software Foundation. The argument itself, at least in the way it has played out on Reddit, blog posts, and mailing lists, is a pretty shabby he-said-she-said spat that should never have seen the light of day.

The lead developer has announced that Libreboot will continue, just no longer under the GNU aegis, in protest of the offense; in response, the FSF has decreed that Libreboot is still a part of GNU. As a lead developer, you can check your project out any time you want, but you can never leave, because GNU is life.

This led me to wonder: Does the FSF have a code of conduct? I know that the

Apache project does, and Django, and Gnome, and tons of conferences do too. But the closest thing the FSF has to a written set of guidelines for its employees to follow to ensure good behavior is a section on the wiki lecturing people on why they're wrong to use the phrase *Open Source* instead of *Free Software*. A code of conduct is pretty useless if you're one man against the world, trying to fight against the evils of proprietary software in the face of huge odds. It's pretty useless if there are just a few of you, all fighting the good fight together. If your team grows, it's tempting to imagine that you're all on the same side and there will never be any conflict that you can't resolve with a cup of tea and an open discussion. But that's the wrong approach.

Codes of conduct aren't just a form of bureaucracy, they exist to protect the organization and the people who comprise that organization. A written code of conduct

gives a framework for disagreements, which stops things from getting personal. People aren't perfect, and they need this sort of structure to keep a cross word, when they're tired and irritable, from escalating into something more serious.

Here's why I think the FSF doesn't have a code of conduct: It would be a reminder that it isn't perfect. Institutions with a moral crusade at their heart often have difficulty compromising, and a code of conduct is all about establishing a framework for compromise. Codes of conduct are about pragmatism, not morality, so they just don't fit the holier-than-thou ethos. If you believe that you're on the side of the angels, you're in the right, and that's the end of the discussion. This attitude, quite rightly, doesn't work in the real world. One day the FSF will join us in the real world instead of polishing its halo, and we'll all be much better off. ■■■

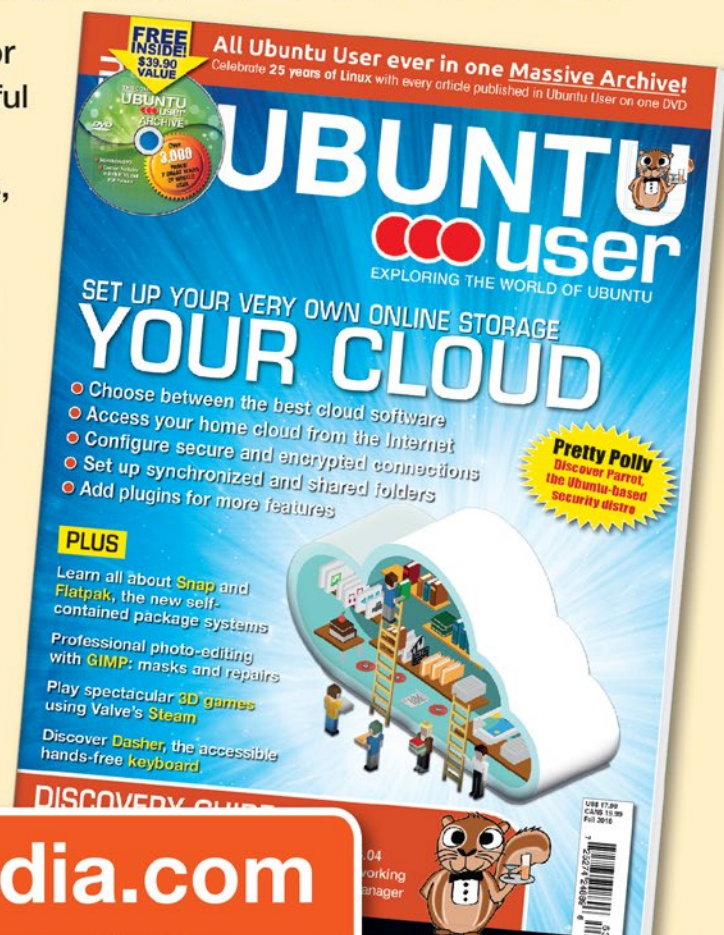
DON'T MISS A SINGLE ISSUE!

The first print magazine created specifically for Ubuntu users! Ease into Ubuntu with the helpful Discovery Guide, or advance your skills with in-depth technical articles, HOW-TOs, reviews, tutorials, and much, much more.

SUBSCRIBE NOW!

4 issues per year for only
£ 24.90 / EUR 29.90 / US\$ 39.95

- ✓ Don't miss a single issue!
- ✓ Huge savings – Save more than 35% off the cover price!
- ✓ Free DVD – Each issue includes a Free DVD!



shop.linuxnewmedia.com

Editor Upgrade

We explore the new goodies in the latest release of this classic text editor.

MIKE SAUNDERS

Forks in the free software world are usually a messy business. Egos get damaged, development effort dilutes, and it takes a long time before the two resulting projects end up working together. But, at the same time, forks are sometimes necessary to keep the bulk of contributors happy and keep the project moving along – look at what happened with XFree86 and X.org, for instance. The former was moving at a snail's pace and failing to attract new developers, but when the X.org fork arrived and most of the XFree86 team moved over, things got a lot better.

What has this got to do with Vim [1], the venerable command-line text editor with its notoriously steep learning curve? Well, Vim has had its own set of problems over the years. It's a very mature and reliable editor, but the code underpinning it was packed with cruft and suffering from bit-rot. This is inevitable given that Vim runs on pretty much every major operating system you can name – but the ancient codebase was impeding progress. New features were hard to implement, and new contributors were sometimes scared off.

So, a fork called Neovim came to life in 2014, with the goal of drastically refactoring Vim's code to provide better scripting support and improved

performance and to generally clean up the cruft. For a while, it looked like Neovim could become the new Vim – just like X.org became the new XFree86. Bram Moolenaar, the lead developer of Vim, was skeptical about some of the changes in Neovim, but he didn't just close up shop and

declare Neovim as the future (Figure 1). No, he and other Vim developers have been incentivized by Neovim to improve the original codebase – so, whatever happens with Neovim in the months and years to come, the fork has definitely benefitted the original project as well.



A Decade of Waiting

Vim 8 was released in September 2016 and is the first major release since Vim 7 way back in 2006. (Although in fairness, versions 7.1 to 7.4 of Vim included some pretty major updates as well.) Given that Vim is one of the most popular text editors in existence, and many developers can't imagine living without it, we thought we'd take a close look at the new release and see what has changed. As you'll discover over the next few pages, many of the updates are "under the hood," rather than user-facing, but the new version has plenty of goodies that will improve your day-to-day text editing life, as well.

Once we've whetted your appetite, you can check your distro's package manager to see if Vim 8 is available. If you want to build the new release directly from source code, head over to the Vim website for downloads and instructions [2] – we provide a quick install guide later in this article.

Five Things to Try

The best new features to speed up and simplify your editing workload.

1 Asynchronous I/O

Previous versions of Vim were very limited in terms of running background processes. If you needed to perform such a task, like running a test suite, Vim would usually wait until the process was completed, meaning a lot of waiting around if you run many such processes regularly. With Vim 8, the editor can run tasks entirely in the background and then exchange messages with them asynchronously using channels.



Figure 1: Bram Moolenaar has led Vim development since the early 1990s. image © <http://tinyurl.com/mj53qzb>.

So, imagine you have a background process that performs some checks on the source code file you're editing. Previously, Vim would wait for the process to complete before showing the results, but now the editor can run the task and communicate with it while you continue editing. If the background task needs to alert you to something, it can tell Vim to display a message, move the cursor to the appropriate place in the file, or do whatever else it deems necessary.

Additionally, Vim 8 supports JSON, the JavaScript Object Notation format commonly used for data interchange. This allows developers to write complex plugins in a variety of languages that then communicate and exchange data with Vim in a commonly used, standardized format.

2 Timers

This is another huge step forward and is related to the previous feature. With timers, you can call a function in Vim after a certain amount of time and have it performed only once or repeated at regular intervals. This may simply be a function built in to Vim, or you could use it to call a background process as described earlier.

So, what could you potentially use this for? Well, on its own it might not sound especially useful, but in the context of Vim scripts, it can do a lot. You could use it to make backups of the currently opened file or regularly add information to a status bar in the editor (such as a clock, or CPU load indicator, or a number of new email messages that have arrived). We expect plugin developers to make big use of this feature in particular.

3 Packages

Some people prefer to use Vim in its most vanilla form, maybe with a few customizations to the `.vimrc` configuration file to make it look a bit prettier. Other users – especially developers – build up a massive armory of plugins and add-ons to turbocharge the editor, but until now, there has been no standard way to manage them. Sure, we have third-party solutions in the form of Pathogen, Vundle, and others, but wouldn't it be better to have this dealt with inside the editor itself?

Well, with Vim 8, native support for packages is now included. You won't need to install an add-on just to be able to install other add-ons – the process will be a lot simpler. At the time of writing, as Vim 8 had only just been released, many popular plugins had not been converted to use the new package format. But, we hope to see some of our favorites moved over to the format soon, including:

- Vim-Airline [3] – An attractive and versatile status bar (Figure 2).
- Solarized [4] – Smart and eye-friendly syntax highlighting schemes (Figure 3).

```

1 function! MyOverride(...)
2   call a:1.add_section('StatusLine' , ' all ')
3   call a:1.add_section('Tag'       , ' your ')
4   call a:1.add_section('Search'   , ' base ')
5   call a:1.add_section('Title'    , ' are ')
6   call a:1.add_section('TabLineSel', ' belong ')
7   call a:1.add_section('ErrorMsg' , ' to ')
8   call a:1.add_section('StatusLineNC' , '%f ')
9   call a:1.split()
10  call a:1.add_section('Error'      , '%%% ')
11  return 1 "tells the pipeline to write the statusline with the builder
12 endfunction
13 call airline#add_statusline_func('MyOverride')
all your base are belong to vim 7%

```

Figure 2: Vim-Airline is one of the most popular Vim plugins, adding an attractive status line at the bottom.

- Surround.vim [5] – Key bindings that work with parentheses, brackets, quotes, XML tags, and so on.
- NERD tree [6] – A tree-form filesystem exploration tool.
- Fugitive.vim [7] – Pretty much essential if you use Git.

4 Window IDs

If you're a casual Vim user but haven't delved into the more advanced features of the editor, you may not be aware of windows – or “viewpoints onto a buffer” in Vim parlance. You can open multiple windows in Vim to view different files or, indeed, work on the same file, and this feature is especially useful if you're not also rocking a terminal multiplexer such as Screen or Tmux.

Anyway, until now, windows in Vim were given numbers for identification, and as you opened and closed windows, these numbers would change. This was pretty frustrating if you remember that you had `/etc/hosts` open in window 5 but, because of some other window operations, that has now moved to window 2. Well, in Vim 8, windows now have their own unique IDs, making them a lot easier to find and work with.

5 Gtk 3 Support

This might not sound like the biggest deal, given that many of us use Vim purely in a terminal window, but for those who prefer an editor that fits in nicely with Gnome or Xfce, this is a welcome change. It's still possible to build Vim with a Gtk 2 interface, and Bram states that the differences between the UIs should be “hardly noticeable.” In any case, it's good to see

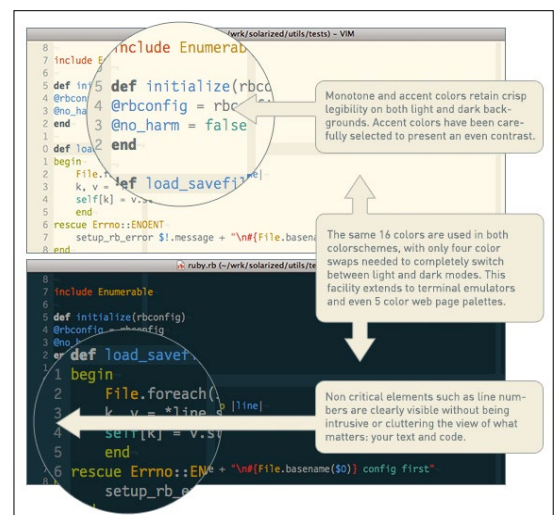


Figure 3: If you find Vim's default color scheme off-putting, try the Solarized plugin.

How to Install Vim 8

You may now be itching to try all these new features and get the shiny new version. Unless you're running a rolling release distro such as Arch, however, you may have to wait a while before Vim 8 packages are available in your repositories (and you may have to upgrade to a new release of your distro). Fear not, though, because it's also not difficult to build Vim from source code. This requires a bit of work at the command line, but if you like (or want to learn) Vim, chances are you're not afraid of the Bash prompt anyway.

First, grab `vim-8.0.tar.bz2` from your nearest mirror [9], and in a terminal window, `cd` into the directory where you downloaded it, and extract it:

```
tar xfv vim-8.0.tar.bz2
```

This will create a new directory called `vim80`, so `cd` into that. Now you can run the configuration and code-building process with one command:

```
make
```

During the configuration stage early on, you'll be alerted to any missing dependencies – the main one you need is the development headers for Ncurses, so install that if you don't have it already (e.g., `sudo apt-get install libncurses-dev`). Once you've dealt with your dependencies, run `make` again, and compilation will begin.

Next, when the source is built and you're returned to the prompt, run this command as root:

```
make install
```

(which is fair enough given how bare it looks), here's a quick guide to help you on your way. Vim has a very steep learning curve, and users still discover new tricks after decades of working with it, but once you have the basics sorted out, you can gradually discover its true power.

The first thing to do is create a simple Vim configuration file. Bare-bones Vim looks pretty intimidating, so put the following text into `.vimrc` in your home directory:

```
set number ruler laststatus=2
hlsearch ignorecase title wildmenu
syntax on
```

Now open that file in Vim (e.g., `vim .vimrc` in a terminal window), and you'll see a more attractive setup than vanilla Vim, with a ruler along the bottom showing the file name, current line, and current column, along with line numbers down the left and syntax highlighting for the configuration file.

Next, start exploring some of the movement keys. Press *H*, *J*, *K*, and *L* and see what they do – they move the cursor around, and may seem redundant when most keyboards have arrow keys. By using *HJKL*, however, you can navigate without having to move your fingers away from the center of your keyboard. Also try pressing *W* to move forward a word and *B* to move back. That's a lot faster, isn't it?

Try opening a bigger file in Vim, and then use `Shift+H-M-L` to move around inside the screen – the top, middle, and bottom, respectively. This is also much faster than pressing keys repeatedly or holding them down for a while. Press lowercase *G* twice (`gg`) to go right to the top of a file, and `Shift+G` to jump to the bottom. Another very useful way to navigate is using curly braces – `{` and `}` – which jump between paragraphs. Very useful when editing lots of text!

Vim works in two modes – command and insert – which takes some getting used to. As you're moving around, you're in command mode, but press `/` to insert text (you'll see "INSERT" at the bottom). When you're done editing text, press `Esc` to quit back to command mode. There are other keys for inserting text, such as `Shift+A` to append to the end of the line and `Shift+O` to open a new line beneath the current one. Always remember to hit `Esc` after editing, so you are in command mode by default, and enter `:w` to save a file and `:q` to quit (`:q!` to quit without saving).

Vim Is a Language

To really grok Vim, you need to think of it as a language, with commands being issued like verbs and objects following. For instance, take the following piece of text:

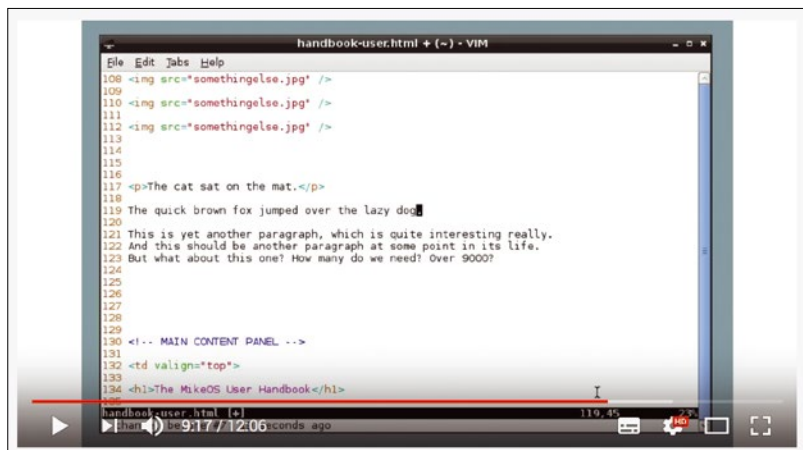
the Vim codebase being updated in terms of the front end as well.

Those are five of the most significant changes in Vim 8, but there's a lot more to discover as well, including partials, lambdas, and closures; a new testing framework for developers; extra options; and even new insert modes. Check out the resources [8] for the full list, including pointers to Help resources that you can use in Vim to explore the new features in greater detail. Your fresh new Vim 8.0 installation should be in `/usr/local`, so enjoy!

Figure 4: To demonstrate why Vim is more than a mere text editor, we made a video [10] – take a look.

New to Vim? Get Started Here

If you've never used Vim before, or you dipped your toes into its waters but ran away scared



```
foobar(baz);
```

Move the cursor into the “baz” part, and then press the following keys exactly as written:

```
ci(hello
```

When you're finished, press Esc. What this means to Vim is: Change the text inside the brackets to hello. Do you see how it's like a verb? And even better, you can repeat these “sentences” by pressing the full stop (.) key. Go down to the next “baz” and press full stop (period), and it performs the whole operation again.

Info

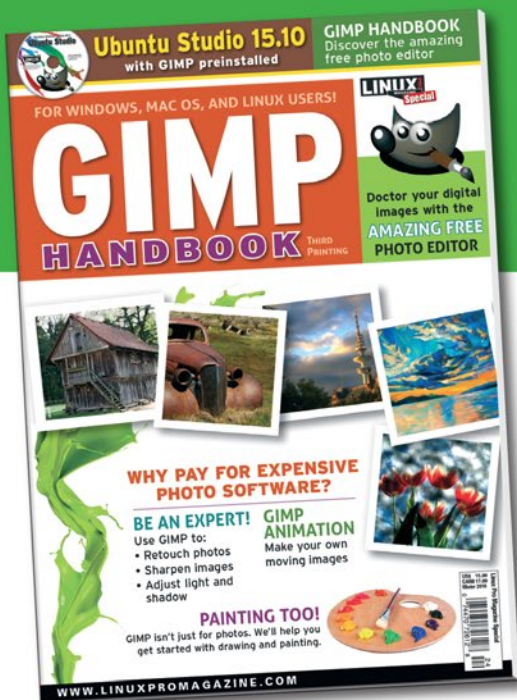
- [1] Vim: <http://www.vim.org/>
- [2] Vim downloads: <http://www.vim.org/download.php#unix>
- [3] Vim-Airline: <https://github.com/vim-airline/vim-airline>
- [4] Solarized: <https://github.com/altercation/vim-colors-solarized>
- [5] Surround.vim: <https://github.com/tpope/vim-surround>
- [6] NERD tree: <https://github.com/scrooloose/nerdtree>
- [7] Fugitive.vim: <https://github.com/tpope/vim-fugitive>
- [8] Resources: <https://github.com/vim/vim/blob/master/runtime/doc/version8.txt>
- [9] Download: <http://www.mirrorservice.org/pub/vim/unix/>
- [10] Video: https://www.youtube.com/watch?v=rf19KQb_HVk

Now, we've only done a very simple editing operation here, but you can already see what's possible if you string together much more complicated text manipulation commands. Once you're familiar with Vim, editing jobs that used to take hours (or required fiddling with scripts or horrible-looking regular expressions) end up taking only seconds. You stop thinking of Vim as a text editor, but rather a powerful command-driven machine for manipulating text.

Those are just the basics – to see what else Vim is capable of, check out the Linux Voice video [10] we made that goes further with the editor's very powerful macro system (Figure 4). ■■■

Shop the Shop

shop.linuxnewmedia.com



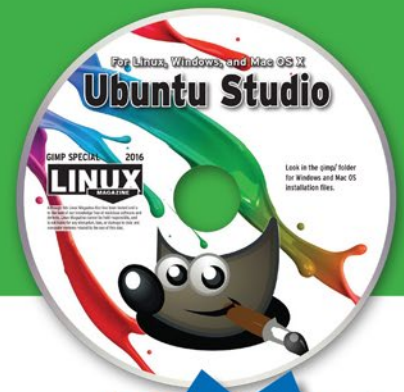
GIMP HANDBOOK

**SURE YOU KNOW
LINUX...**

but do you know GIMP?

- Fix your digital photos
- Create animations
- Build posters, signs, and logos

**Order now and become an expert
in one of the most important and
practical open source tools!**



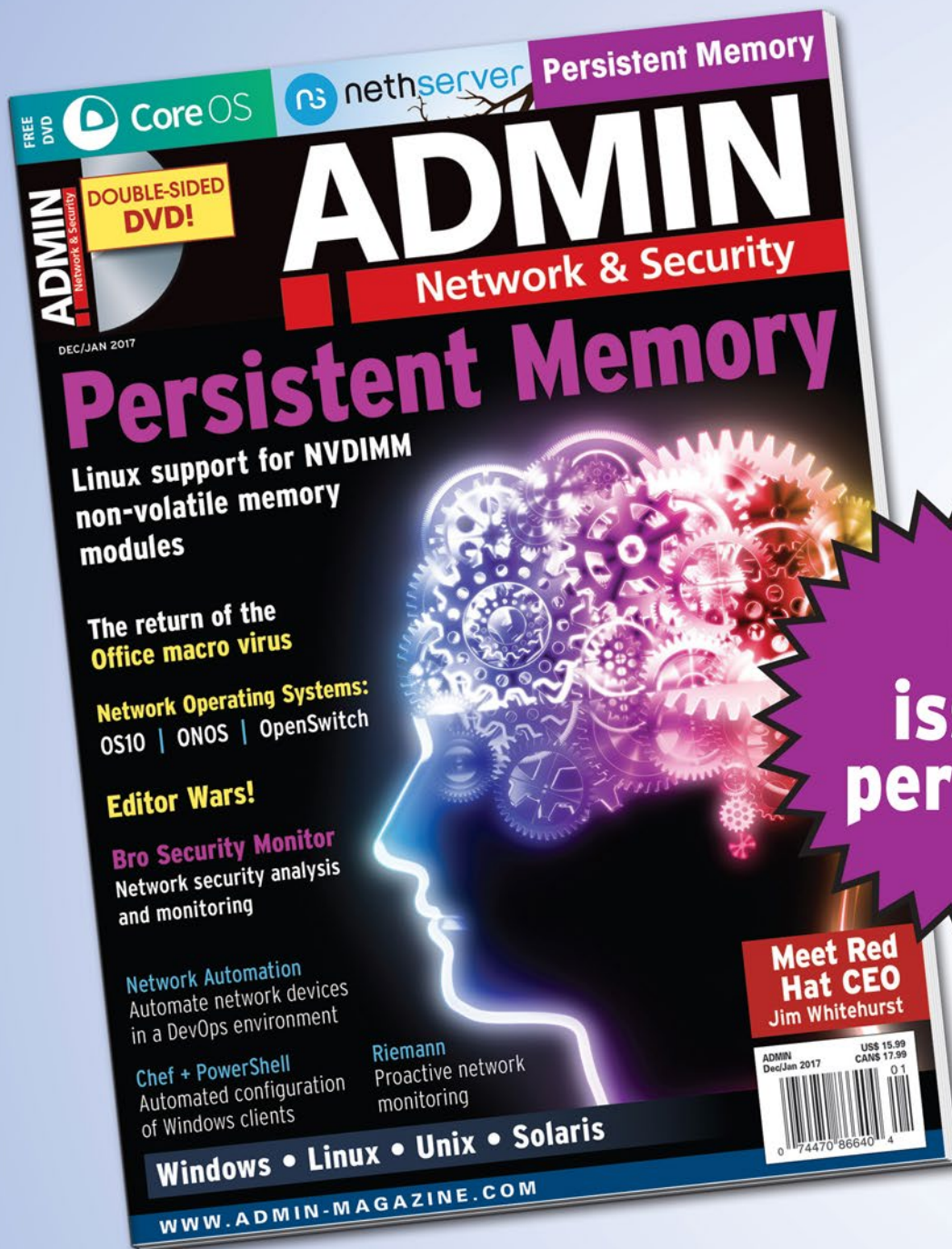
GIMP
Doctor your digital
images with the
**AMAZING FREE
PHOTO EDITOR!**

Order online: shop.linuxnewmedia.com/specials



FOR WINDOWS, MAC OS, AND LINUX USERS!

SUBSCRIBE NOW



6
issues
per year!

The New IT

New tools, new threats, new technologies...Looking for a guide to the changing world of system administration?

shop.linuxnewmedia.com

AND SAVE 30%

Explore the new world of system administration

It isn't all Windows anymore - and it isn't all Linux. A router is more than a router.

A storage device is more than a disk. And the potential intruder who is looking for a way around your security system might have some tricks that even you don't know. Keep your network tuned and ready for the challenges with the one magazine that is all for admins.

Each issue delivers technical solutions to the real-world problems you face every day. Learn the latest techniques for better:

- network security
- system management
- troubleshooting
- performance tuning
- virtualization
- cloud computing

on Windows, Linux, Solaris, and popular varieties of Unix.

**REAL-WORLD
PROBLEMS
SOLVED!**

ADMIN
Network & Security

shop.linuxnewmedia.com

FAQ

NoSQL

Big data is like The Matrix – Better without the sequel

Ben Everard

Q Right, before I understand what NoSQL is, can you give me a quick rundown on SQL?

A That's a great place to start! SQL (or Structured Query Language, often pronounced sequel) is the standard method of getting data out of databases. Its popularity is shown by the fact that it's in almost every major database name: MySQL, PostgreSQL, SQLServer, Oracle....

Q Hang on, there's no SQL in the name Oracle.

A OK, the name thing doesn't work with every database, but the Oracle database is still based around SQL as is MariaDB.

The language is emblematic of the relational style of databases that's dominated the data-storage industry pretty much since there have been computers with enough storage to call themselves databases. The idea behind relational databases is that information can be mapped to tables, and these tables can be linked to create complex data models.

For example, if you ran a shop, you could have a database with a table of stock. Each line in the table would be about one item that you stocked and include things like the number of items

you currently had. You could also have a table for suppliers that included their address, payment information, etc. The stock table could include a supplier reference number that you could look up in the supplier table. When you needed information from the database, your software could combine these two tables (known as joining them) so that you got a view with each item and the details about the supplier of that item.

One of the great features about the relational model is that each bit of information is only stored once. This is great for data integrity because it means that when a piece of data changes, you only have to update it in one place. If, in our previous example, a supplier changed their address, you'd only have to update the supplier table, and then every query would pull out the right result.

Q SQL's sounding pretty good so far. What has anyone got against it?

A Not many people are really against SQL, but there are some classes of information where it's not necessarily the best fit.

SQL databases were designed as ultra-reliable stores for important data, and they still fit this

role well. However the model comes with some overheads, and when you start to deal with really large volumes of data, these overheads can get significant. There is a wide range of NoSQL databases, so I don't want to generalize, but typically, they work best when you've got a large amount of data to store and want it spread out across a lot of machines to process it.

Q Can you give me any examples of when NoSQL works well?

A One that springs to mind is the ELK (Elasticsearch, Logstash, Kibana) stack for monitoring machines. This is typically set up when you have a large number of machines that you want to keep an eye on. They send all their logs to a central Logstash server that processes them and puts the log information into an Elasticsearch database. You might also gather other information, such as the CPU usage or free memory statistics and push them all into the data store as well. Kibana is a web front end that can provide real-time visualizations on what the data looks like.

You can push lots of data in even if it's in different formats – you could have Apache logs in the same store as system logs and Elasticsearch wouldn't care. It

leaves the task of decoding the information to the user rather than trying to encapsulate it in the structure of the tables.

In this setup, you can very quickly end up with very large amounts of data, yet at the same time, you want to be able to process it very quickly. Once the logs are written, they never change. Rather than all the guarantees about data consistency that made SQL databases great for their style of data stores, what's really important in an ELK stack is speed. Any delays in writing data to the database mean more processor resources are needed. Likewise, complex visualizations need to be generated in almost real time.

Q OK, so NoSQL allows you to store a wide variety of differently organized data in the same place and get it back quickly?

A Well, yes and no. In the previous example, we looked at Elasticsearch, which is a document-orientated database, meaning that it's a place you can keep pushing records in any format (which are the documents). This is one of the most common types of NoSQL database, and many of the most famous new databases work in this way, such as MongoDB and Couchbase. However, others work in very different ways. For example, key-value stores simply allow you to store data that can be retrieved using a key (unlike other types of database, the records can only be accessed by the key and not by the other data in the record). These key-value stores (e.g., Riak, Memcached, and Apache Cassandra) are highly scalable and perform well even under heavy load. There are also graph databases for storing highly connected data. There isn't a set group of database types that classify as NoSQL; it's just anything that's not relational. One big difference between SQL and NoSQL databases is that

NoSQL databases sometimes can have slight inconsistencies in data.

Q Inconsistencies in data? That sounds pretty bad! What's the point in a database if the data isn't correct?

A Relational SQL databases are usually designed around the ACID principal. That is, each action is Atomic (either the entire action happens or none of it does), Consistent (after every action, the entire database is correct), Isolated (only one action happens at a time) and Durable (actions are permanent even if there are failures). This means that whatever you do with the database, nothing can ever be wrong with the data.

For example, you can't end up reading the wrong data because a write operation is half-way through, or a broken write operation can get half the data in but not the other half.

NoSQL databases are often designed around the BASE principal. That is, Basic Availability (the database keeps running), Soft State (the data can change at any time because actions aren't atomic) and Eventual Consistency (at some point in the future the database will be right, but it may serve slightly stale data at some points).

BASE guarantees are quite weak, but they're fine for occasions where it's OK for the result of a query to sometimes be a second or two old (such as in a messaging application), and these weaker guarantees make it much easier to scale the databases to huge numbers of users. However, if you need to always get the correct and most up-to-date information with no exceptions, you need a database with ACID guarantees.

Q So if you don't use SQL, what language do you use to query these databases?

A There's no standard, and each different database has developed their own language. Some have created new

query languages from scratch, and others have built on top of existing programming languages (e.g., JavaScript in MongoDB). Several NoSQL databases can be queried in SQL.

Q Wait, hang on a minute. I thought NoSQL meant that there was, well, no SQL.

A Well, it once did – and for many people it still does. However, some projects have put SQL query layers on top of non-relational databases to make them more approachable for people who are familiar with this language. NoSQL is now held by some to mean not only SQL. In truth, the name NoSQL has always been a misnomer because the key factor for this type of database isn't the query language that they use but the fact that they aren't relational.

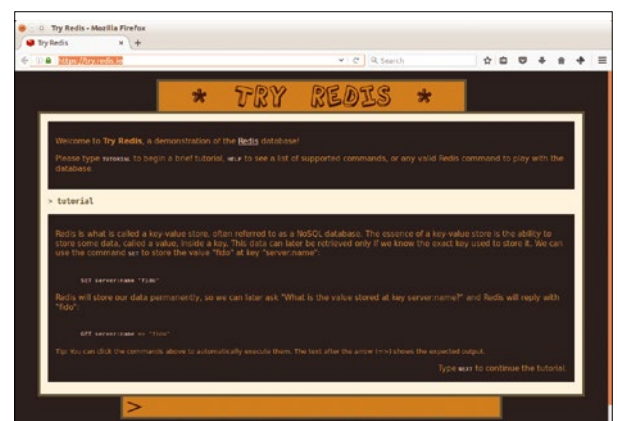
Q This all sounds really interesting. How can I try out NoSQL?

A As NoSQL is diverse class of databases, it's hard to give a single place to start. A good place to dip your toe in the water is the Redis online demonstration [1], which allows you to interact with this key-value store via your web browser (Figure 1). Just don't forget that this is only one type of one class of NoSQL database and that they all work in different ways. ■■■

Info

[1] Try Redis: <https://try.redis.io/>

Figure 1: NoSQL with NoInstalling – try the Redis key-value store database online.





Valentine Sinitsyn develops high-loaded services and teaches students completely unrelated subjects. He also has a KDE developer account that he's never really used.

CORE TECHNOLOGY

Prise the back off Linux and find out what really makes it tick.

One ip tool to rule them all

When it comes to network configuration, Linux has several utilities collected in *net-tools*. Users learn to manage addresses with `ifconfig`, routes with `route`, and MAC addresses and the local network segment (the neighborhood) with `arp`. A single tool, `ip` of the *iproute2* tool collection, replaces several of the classic network tools with one utility.

Making Links

The `ip` tool operates on objects, which could be links, network layer addresses, routes, rules, and a few others. I'll start with Layer 2 objects (i.e., links) and advance to upper layers.

The `ip` utility should already be in your Linux system. If not, install the *iproute2* package. The `ip(8)` man page provides generic instructions on using the command, whereas `ip-link(8)` and friends provide the specifics. If you have ever used Git, you understand this split. Basically, you provide `ip` an object on which to operate; a command, such as `add` or `del`; and some options. The command `ip <object> help` lists the details in each case.

Before you can use a networking interface in Linux, you need to bring it up. The following command sets up a network interface:

```
# ip link set up dev eth0
```

This command must run as root (note the `#` prompt). The `link` object refers to a networking device, either physical or virtual, and the `set` command sets various link options. For instance, you could enable promiscuous mode (useful for sniffers such as Wireshark) with `ip set promisc on` or rename the interface with `ip set <name>`. The `up` option tells `ip` you want to set the link active, and `dev eth0` specifies the device. You can check whether the command worked as expected with `ip link show` (Listing 1).

In the preceding command, the `show` command doesn't require root permissions. If you omit it altogether (i.e., `ip link`), `show` is implied; adding the `-s` switch collects and shows a few statistics (Figure 1). The `UP` flag in angle brackets tells you the link is up, and `NO-CARRIER` indicates that no cable is currently plugged in to the laptop. Note that MAC address is displayed as well, which you can change with:

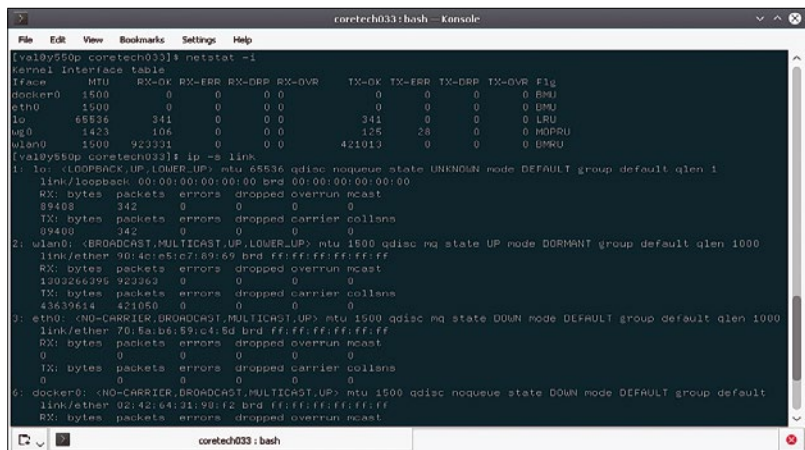
```
ip link set address
```

In addition to configuring existing network links, `ip` can create new links – or at least a new virtual Ethernet (Veth) adapter:

```
# ip link add veth0 type veth peer name veth1
```

Veth interfaces come in pairs and are much like real network cards connected with a patch cord. Entering `ip link show` displays them as `veth0@veth1` and vice versa. Veth interfaces were designed to simplify network communication for namespaces, and they are often seen with containerization tools such as Docker, but you can also use a virtual interface in other situations where you just want a simple network tunnel. The command `ip link add` can also create bridges (acting as a `brctl` substitute), VLANs, and a variety of network tunnels, including the VXLAN and Geneve options commonly used in network virtualization. You'll hardly ever need all these link types on your machine, but if you did, `ip` could manage them.

Figure 1: `ip` gathers stats and can even substitute for the venerable `netstat`, to some extent.



If you no longer need a link for some virtual device, use `ip link del` to get rid of it, but don't expect this command to work for physical devices.

Meeting Neighbors

Up the stack, you find the Network layer (aka, Layer 3) and network addresses. IPv4 is a typical specimen. The IPv4 address is a bit of information you usually supply when you bring the interface up with `ifconfig`; `ip` supports both IPv4 and IPv6.

Your single entry point to all address-related operations, regardless of address type, is `ip address`. Adding an address is easy:

```
# ip addr add 192.168.1.2/24 dev eth0
```

Look at how `192.168.1.2/24` specifies both the IPv4 address (`192.168.1.2`) and the netmask (`24`). This information is already enough for `ip` to deduce a broadcast address, which is often set explicitly with `ifconfig`.

It is also completely possible to assign an interface more than one address (Listing 2). Note that `ip` reuses the same “verbs” (or commands) across different objects.

Older tools (like `ifconfig`) provide multiple address support via interface aliases (`eth0:0`). For `ip`, this is not a requirement. Yet you can achieve the same effect if you assign an address label as in Listing 3.

For backward compatibility, the `label` string must start with the interface name followed by a colon. You may delete addresses you no longer need with `ip address delete` (or just `ip a d` – see the “Abbreviations and Synonyms” box), whereas `ip address flush` flushes all addresses assigned to the interface.

Once you have assigned an interface a new IP address, how do other computers know about it? Neighborhood protocols are the answer. For IPv4, it's ARP (LV031); IPv6 calls the equivalent mechanism Neighborhood Discovery (ND), which is a part of ICMPv6. `ip` handles both ARP and IPv6 ND via the `neighbour` object. In practice, `ip neigh` is the prevalent command form.

The most common operation is to list ARP/ND entries. The old-school `arp` command lists IPv4 addresses only, yet `ip` happily handles both families (Listing 4).

You can instruct `ip` to show IPv4 or IPv6 entries only with the `-4` and `-6` switches. You can also add new neighbor entries and change, replace, or delete existing ones. The `ip-neighbour(8)` man page covers the necessary commands.

Finding Routes

Network layer protocols (IPv4 or IPv6) are all about routing. Routing is how packets reach their destinations beyond the local network segment. In the simplest case, routing information boils

Listing 1: Check a Networking Device

```
$ ip link show dev eth0
2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast
state DOWN mode DEFAULT group default qlen 1000
link/ether ec:f4:bb:29:87:3d brd ff:ff:ff:ff:ff:ff
```

Listing 2: Assign More than One Address

```
# ip addr add 192.168.1.3/24 dev eth0
# ip addr show dev eth0
2: eth0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast
state DOWN group default qlen 1000
link/ether ec:f4:bb:29:87:3d brd ff:ff:ff:ff:ff:ff
inet 192.168.1.2/24 scope global eth0
valid_lft forever preferred_lft forever
inet 192.168.1.3/24 scope global secondary eth0
valid_lft forever preferred_lft forever
```

Listing 3: Assign an Alias

```
# ip addr add 192.168.1.4/24 dev eth0 label eth0:0
# ifconfig eth0:0
eth0:0 Link encap:Ethernet HWaddr ec:f4:bb:29:87:3d
inet addr:192.168.1.3 Bcast:0.0.0.0 Mask:255.255.255.0
...
```

down to the default gateway address, which accepts all packets destined for the outside world. In a large enterprise or provider network, complex dynamic routing protocols are employed to ensure connectivity and optimize traffic flow.

Although `ip` doesn't do anything with dynamic routing protocols – separate daemons are in charge of this – it does provide all you might want for your static routing configuration, even in advanced scenarios. In a typical case, the route is chosen on the basis of the destination address, and that's the only use case the `route` command covers. Not only does `ip` cover this basic scenario, it provides a few additional options, as well:

```
# ip route add 192.168.2.0/24 via 192.168.1.1
```

This command adds a route to the `192.168.2.0` network via the `192.168.1.1` gateway. A special keyword, `default`, which is equivalent to `0.0.0.0/0`, can be used to specify the default gateway. You can delete a route with `ip route delete` and modify existing routes with `ip route modify`. As usual `ip route show` (or `list`) displays the routing table:

```
$ ip route list
default via 192.168.101.1 dev wlan0 proto static 2
metric 600
...
```

For a complex routing table, it might not be evident which route will apply to which destination,

Listing 4: List ARP/NC Entries

```
$ ip neigh show
192.168.101.1 dev wlan0 lladdr 04:8d:38:c2:88:b3 STALE
2a02:6b8:0:c33::1 dev tap0 lladdr 00:bd:70:48:00:00 router REACHABLE
...
```

Listing 5: Local Table 255

```
$ ip route list table local
...
local 192.168.101.43 dev wlan0 proto kernel scope host src 192.168.101.43
broadcast 192.168.101.255 dev wlan0 proto kernel scope link src 192.168.101.43
```

Abbreviations and Synonyms

Two things to note: First, `ip` supports abbreviations. There is no minimum length, as long as the shortcut remains unambiguous. Entering `ip addr` is the same as entering `ip a`. Second, commands may also have synonyms: `list` and `lst` are the same as `show`.

especially if routes involve different `metric` or `preference` values. The `ip route get` comes to the rescue: It runs the virtual packet through the routing table and prints the result. No root privileges are required:

```
$ ip route get 8.8.8.8
8.8.8.8 via 192.168.101.1 dev wlan0 src 192.168.101.43
cache
```

`ip route get` resolves the route you provide, creating so-called clones (think cache entries) as necessary. The word `cache` in the last line signifies this fact. One can see what's currently in the routing cache with `ip route list cache`, and flush it with `ip route flush cache`. However, the IPv4 routing cache was removed in Linux 3.6 for its non-deterministic performance. So, unless you use an old kernel, `ip route list cache` yields nothing. In a nutshell, the kernel still makes route clones but doesn't try to store them anymore.

The route type shown above is the most typical one, but it is not the only one `ip` understands. `iptables` was the traditional way to silently discard packets going to evil networks, but the `blackhole` route would also work fine:

```
$ ip route add blackhole 8.8.8.8
# ping 8.8.8.8
connect: Invalid argument
```

For local sockets, the `blackhole` route yields the `EINVAL` error. Two similar route types, `unreachable` and `prohibit`, generate `ICMP Host Unreachable` and

`ICMP Communication Prohibited` replies. Local senders get `EHOSTUNREACH` and `EACCESS`, respectively.

A throw route type terminates lookup in the current routing table. Is there

more than one routing table in Linux, you ask? That's a good question!

Play by the Rules

Since version 2.x, Linux has supported multiple routing tables – around 2^{31} of them (which is quite a few), although table numbers 0 and 253-255 are reserved. By default, routes end up in table 254, which is also called `main` (see `/etc/iproute2/rtables`). Table 255 (`local`) contains routes to local and broadcast addresses. The kernel maintains this table automatically (Listing 5).

Note how you tell `ip` the table you want to use. A special value, `all`, lists all route entries. Other `ip route` commands, such as `add` or `del`, support this syntax as well.

Multiple tables are used for policy-based routing, in which the criteria could be pretty much anything: a source address, ingress (incoming) or egress (outgoing) interface, TCP or UDP ports, or even the package payload.

Linux implements policy-based routing via a set of rules, also known as the Routing Policy Database (RPDB), and `rule` is an object in `ip`. The following is the default ruleset on a system with only classical routing configured:

```
$ ip rule show
0: from all lookup local
32766: from all lookup main
32767: from all lookup default
```

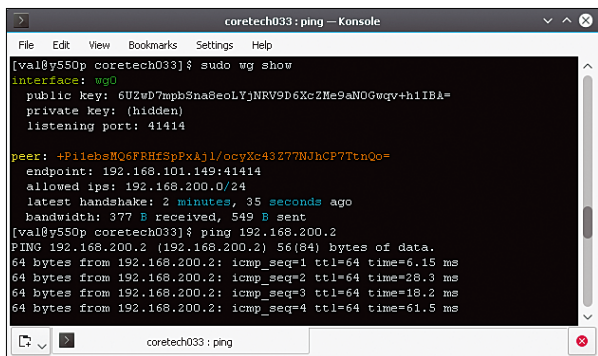
You see that each rule comprises a priority (32766), a selector (`from all`), and an action (`lookup main`). The kernel traverses rules from smaller to larger priority values, so the `local` table comes into play first. The `from all` selector matches any source address. Selectors may also include destination addresses (`to`), ingress and egress interface names (`if` and `oif`, respectively), and firewall marks (`fwmark`). You may assign this mark in your `iptables` rules with `MARK/CONN-MARK` targets. `Iptables` implements sophisticated packet matches (LV017), including Deep Packet Inspection (DPI), and, with `fwmark`, you can route traffic according to these matches. This opens a whole new set of possibilities. A how-to [1] provides some useful hints.

Action is usually a table lookup. By now, you probably get the idea behind policy-based routing. Just mark traffic the way you want, create a routing table per traffic type, and add a rule to look up these tables. Do you want a separate ISP connection for your online gaming needs? No problem!

Building Private Networks

As you see now, `ip` can cope with a wide range of network configuration tasks. Beyond what you've already seen, `ip` can handle multicast, create network

Figure 2: WireGuard's wg tool reporting successful tunnel negotiation. Colorful output is a bonus.



tunnels, and manage TUN/TAP devices. In fact, `ip` is sophisticated enough to pave the foundation for a full-featured VPN solution.

Perhaps the most popular VPN technology in Linux is OpenVPN [2], which creates a pair of TUN or TAP virtual adapters, depending on the settings, and wraps all packets coming through these interfaces in encrypted UDP datagrams. OpenVPN does all network-related heavy lifting itself, and there is nothing wrong with this approach, except it opens the way for simplification.

A recent Linux VPN solution called WireGuard [3] offloads networking to `ip` and focuses on the cryptography. This helps keep its codebase small and clear. WireGuard is under heavy development, so it might not have made its way into your distribution's repositories yet. If this is the case, you can compile it yourself: The homepage has a step-by-step guide.

To set up a tunnel, use `ip` to create `wireguard`-type devices on both peers and assign them IP addresses. This is similar to the Veth example:

```
# ip link add dev wg0 type wireguard
# ip address add dev wg0 192.168.200.1 peer 2
192.168.200.2
```

Note that `ip addr` specifies a peer, because the WireGuard tunnel is a point-to-point link.

Now you generate keys to encrypt the traffic. Although you have several options, for simplicity, you can follow the recipe in the `wg(8)` man page,

```
$ umask 077
$ wg genkey | tee private.key | wg pubkey > public.key
```

where `wg` is the WireGuard userspace tool. Run it on both peers, and create configuration files – for example:

```
[Interface]
PrivateKey = ...
ListenPort = 41414

[Peer]
PublicKey = ...
Endpoint = 192.168.101.149:41414
AllowedIPs = 192.168.200.0/24
```

Set `PrivateKey` to this host's `private.key` contents, and `PublicKey` to the peer's `public.key`. Assign `Endpoint` to the peer's IP address, and apply the configuration with

```
wg setconf wg0 sample.conf
```

on both ends.

To activate the link, you use `ip` once again. Peers negotiate the connection when you bring the link up:

```
# ip link set up dev wg0
```

If everything is fine, running `wg show` should bring the result, as shown in Figure 2.

Splitting cryptography and network-related tasks between `wg` and `ip` helped WireGuard focus on security rather than reinventing the networking wheel. One day the program will reach a 1.0 milestone, and I'll probably allot it a Core Tech article of its own. ■■■

Linux Voice Pro Tip

Most `ip` objects understand `save` and `restore` commands to serialize and deserialize the configuration. Should you want some persistence, they are the way to go.

Info

- [1] The Linux Advanced Routing & Traffic Control HOWTO: <http://lartc.org/howto/>
- [2] OpenVPN: <http://www.openvpn.net>
- [3] WireGuard: <http://www.wireguard.io>

Command of the Month: `ip netns`

This month, the whole Core Tech is devoted to a single command. It would be weird to nominate a different command as the Command of the Month, so instead, I'll honor one of the `ip` objects that is powerful enough to have a tool of its own.

The name of this object is `netns`, and it stands for "network namespaces." The Linux namespaces isolation technique is one of the standing pillars behind containerization tools like Docker. However, you can use network namespaces on their own without a fancy tool like Docker. Network namespaces come in handy whenever you need a separate network stack for some task, even if complete isolation is not a goal.

To create a namespace, use:

```
ip netns add <name>
```

Now, you can configure networking as usual, provided you add `-netns <name>` (or just `-n <name>`) to each `ip` command invocation. In fact, `ip` can execute arbitrary commands (e.g., `ping`) within the namespace with:

```
ip netns exec <name> <command>
```

and `-netns` is just a shortcut for:

```
ip netns exec <name> ip [args]
```

Each namespace comes with its own set of network interfaces, and you can assign a link to the namespace with:

```
ip link set netns <name>
```

The `ip monitor` command reports namespace creation and deletion events, and `ip netns list` shows namespaces defined in the system. If you no longer need a namespace, use `ip netns del` to delete it. Note the namespace might stay alive if it has users other than you. When the last user deletes the namespace, its physical devices are reassigned back to the default. ■■■

FOSSpicks

Sparkling gems and new releases from the world of Free and Open Source Software



Graham Morrison tears himself away from updating Arch Linux to search for the best new free software.

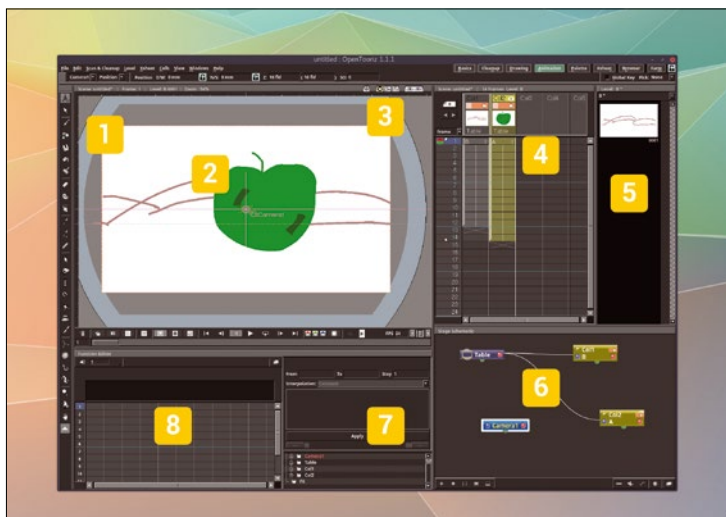
Animation Tool

OpenToonz 1.1.1

The open sourcing of this cartoon-drawing application was a huge story because it was partly based on a piece of software called Toonz. Toonz was used and customized by Studio Ghibli over many years as it produced its monumental catalog of genre-defining movies. If you've not seen anything by Studio Ghibli, you have to watch both *Spirited Away* and *My Neighbor Totoro* as this month's homework. Which is why it was so remarkable when the company released its software as open source. It didn't matter that

this might be because it was bought by a larger company or because Studio Ghibli wanted to own the standard tool that artists are going to want. The result was a best-in-class animation tool that was free to use and free to fork and modify. The team responsible for the Linux conversion is also due a huge amount of credit, because without them, there wouldn't be a Linux version. Only Windows and OS X were supported by the original release, and it's taken some serious code refactoring to get the Linux version near a stable release.

We've been watching development for a while, and it's only over the last month or so that we'd recommend finally trying to get hold of OpenToonz and running it. We built the current version from source, but you should find pre-built packages as its stability continues to improve. You also need to be prepared for quite a learning curve. Rather than what some might assume will be the simplicity of an application like Deluxe Paint, OpenToonz is a seriously professional proposition. That means there are very few concessions to usability, with a user interface geared toward creative freedom and flexibility. However, you can go a long way by thinking of the user interface as a two-dimensional version of Gimp. Time goes down the Y-axis and layers go from left to right along the X-axis. Illustrations can be scanned or drawn and placed into a cell in this grid and animated through the use of an onion skin with each new row. OpenToonz provides lots of plugins and effects for processing and manipulating not just the original images, but how they transition over time – just as you can manipulate video in a video editor, for example, only with clips going down rather than across. You can also perform lots of other functions. The way the images, cells and camera are abstracted means you always maintain maximum quality, which is the most important feature when you're printing your animation to 35mm film for projection onto the cinema screen. In fact, everything except the stability of this application feels professional, from the learning curve, to the plugins, to the sheer complexity of the many dialogs. But don't let that put you off. Stability is improving every week, and there's already an avid community willing to help each other get the most out of this fantastic tool. ■■■



1 Tool palette. Works just like the palette in Inkscape. **2 Image editor.** Preview your animation and draw onto the screen. **3 Model selector.** Different modes let you dive into different details of your project. **4 Level manager.** Just like video editing, only with vertical tracks and more keyframes. **5 Level preview.** Levels are separate layer elements that can move independently. **6 Project schematic.** See how everything is linked together. **7 Interpolation.** Dynamically control movement between frames. **8 Function editor.** Create mathematical curves for smooth movement.

Project Website

<https://blackflux.com/node/11>

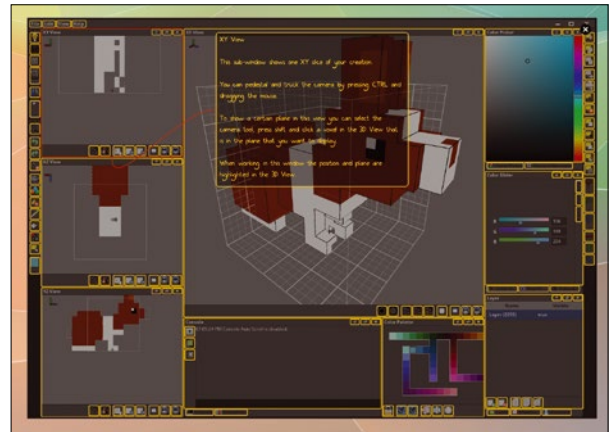
3D Retro Art

VoxelShop

None of us around in the 1980s would have thought that the art style of those pixelated sprites would survive 30 years. Creating something that worked in a 16x16 grid of eight colors was a necessity, not a creative decision. And yet, pixel art is thriving in an era where 400dpi screens are common. This is because the aesthetic of pixel art is difficult to replace by simply creating a higher resolution image. Even when pixelated games are now rendering outside the boundary of simple matrices of color, it's the style and creative limits that remain, making it easier to understand why "voxels" have become so popular. These three-dimensional pixels started off as the building blocks of 3D gaming, before graphical processors could

accelerate triangles; however, like two-dimensional pixels, they've become a creative genre in their own right, mostly thanks to Minecraft.

This also means there's an insatiable appetite for creating 3D voxel art, and that's exactly what VoxelShop does. It's like a 3D sprite editor. Instead of a single matrix of pixels, your creations have layers that can then be rendered from any angle or imported into anything that uses voxels. Some examples have even been sent to 3D printers to get these little models onto your real desktops. This brilliant application even lets you draw 2D sprite art in flat X, Y, and Z squares while rendering the result into the 3D view. You can choose which slice to edit or draw directly into the movable 3D viewport, and you're obvi-



In VoxelShop, you can enable a brilliant help view that overlays the GUI with simple instructions on what everything does.

ously no longer constrained to small grids. Voxels can be any size you like, and there's a neat 3D fill tool that helps you avoid repetitively clicking your mouse. All we really want is virtual reality headset support, but that's not going to happen until either Valve or Oculus port their tools to Linux. ■■■

Project Website

<https://blackflux.com/node/11>

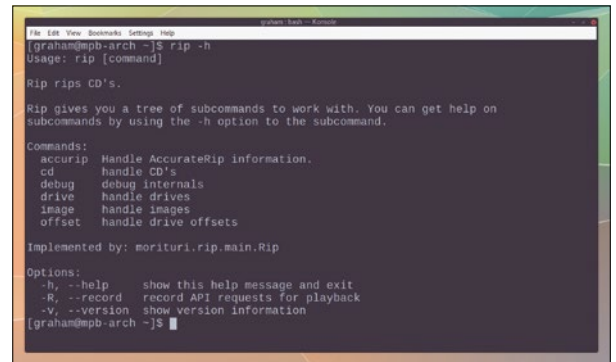
Music Ripper

Whipper

Many of us have found ourselves transported into a world of streamed music with huge collections of physical music products in tow, often collected over decades. For people who love music, streaming services are amazing, often letting you discover new music every day, but the audio quality and breadth of content can seldom compete with your compact disc collection. The solution is to rip these audio discs onto your hard drive, perhaps even storing them online to stream back through services like Google Play Music (although the quality suffers). And yet, getting the audio off your collection isn't always straightforward because, unlike discs holding pure data, it's very easy for an audio rip

to introduce errors as the various frames are read off the disc, as well as jitter as the timing of the rip changes. Of course, audio CDs are also likely to have scratches and blemishes that affect the rip performance, and ripping software needs to be able to restore any missing data as transparently as possible.

This is why, despite this ripping process being well established, a sizeable audiophile subculture has built up around the vagaries of ripping, with ripping software being updated to add the newest methods and ideas. Whipper is a fork of one of the most established rippers, "the morituri project," a tool that has always targeted ripping accuracy over speed. This is what most of us want if we're going to the trouble



Whipper supports modern metadata searches (e.g., of MusicBrainz) and tests the validity of a rip with AccurateRip.

of ripping 500 or 1,000 CDs from a collection, although you obviously need a lot of patience. Whipper can test and copy a rip, detect hidden tracks, use templates for filenames and directories, and create perfect copies of your music, and it's great that this kind of software is still being developed. ■■■

Project Website

<https://github.com/JoelLametta/whipper>

3D Data Processing

CloudCompare 2.7

We've recently become slightly obsessed with photogrammetry. This is the magical process that turns dozens or hundreds of flat ordinary photos taken of a singular object or location and turns them into fully fleshed-out 3D models. And it really works. Clever algorithms detect the same three-dimensional positions in your photos, and use maps of these positions to work out the relative three dimensional positions of other points within the scene, often generating millions of these points of reference. You then take this "point cloud" and triangulate them into polygons before eventually using more clever algorithms to cut and translate textures out of the original photos and map them onto the polygons, creating the 3D model. You

can then use these models within software like Blender, or even in cutting edge technologies like virtual reality, to visit and walk around geographical locations. Rather than sending someone a postcard, you can now send them a fully immersive three-dimensional model of your environment.

CloudCompare is a fundamental part of this process. The tool takes these clouds of point data and turns them into polygons. These clouds of points can come from anywhere – a laser scanner, your own mouse, or photogrammetry software (we'd recommend VisualSFM, for this, or the Python Photogrammetry Toolbox). But doing something meaningful with this point data is a tough job. An image we were working with recently had 17 million of these points, generated

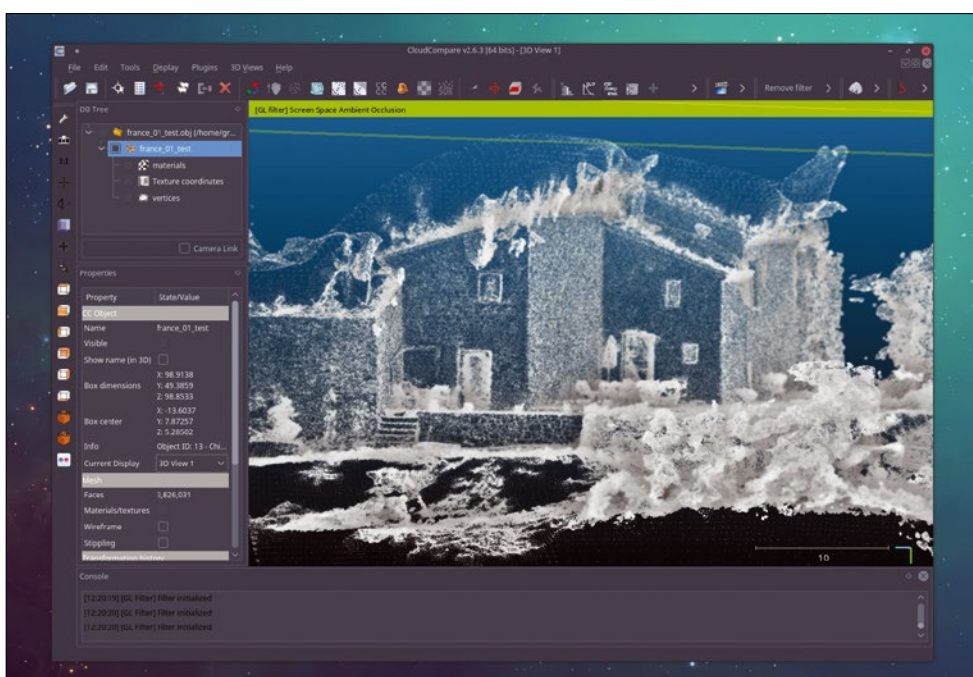
from 700 photographs, which we turned into an OBJ file for Blender with 4 million triangles. CloudCompare accomplished this using half of our 32GB of RAM, which is something paid-for competitors can't do with the same data set (we tried). You have complete freedom over the end model, and most common export formats are supported. Our 3D surfaces were generated using one of the plugins bundled with CloudCompare, but other plugins can be used to create point data from an Xbox Kinect device, detect shapes, and change the OpenGL rendering shaders. New plugins can render animations, perform boolean operations on meshes, and extract the bare earth from LIDAR point clouds. Perfect if you're flying drone cameras above ancient monuments. CloudCompare can do other things with the points too, such as map their density, produce statistics, and change their projections – functions that belie its roots as a change-detecting scientific tool developed by Telecom ParisTech and the R&D division of EDF.

With a few exceptions, notably Blender, 3D software is expensive. That CloudCompare is open source and yet so comprehensively powerful, to the extent that its developers could easily charge serious money for a license, is remarkable. The project does accept donations, but we're seriously impressed that you can build software like this yourself. It may be a little on the niche side, but having such an efficient and powerful tool at our fingertips is exactly why we like Linux so much and, of course, open source. Grab yourself a camera and start scanning your environment! ■■■

Project Website

<http://www.cloudcompare.org>

Rather than sending someone a postcard, you can now send them a fully immersive three-dimensional model of your environment.



Don't worry, CloudCompare has nothing to do with OpenStack, Docker, or Kubernetes.

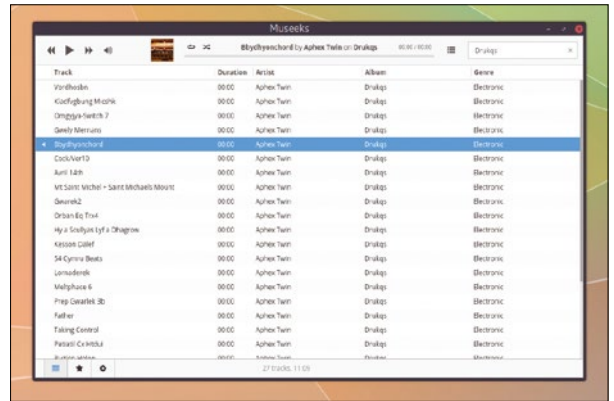
Music Player

Museeks 0.7

We're not sure how we feel about the proliferation of applications written using Node.js, Electron, and React.js. On the one hand, it's great that cross-platform applications can now include Linux with no extra work required. It's something we wish Adobe would consider, for instance. But tools using these JavaScript libraries always feel so bloated, even if the applications themselves are excellent (which they often are). If the 120MB installation size of the music player Museeks doesn't put you off, especially if you like to work with the same application across different operating systems, it's definitely worth a try. The user interface is minimalist and won't even use your window manager's sur-

round unless you enable it. This is a good thing, and we like the design. It doesn't stop you getting started either, which is as simple as adding a new source to your library and refreshing the internal database. You can have multiple folders, too.

With music added to your library, every available track is listed within the main view. There's no artist or album hierarchy, nor is the list sorted by folder groups. This isn't going to work if you have a large collection of music, but it's simple, uncluttered interface doesn't leave you with any distractions. Audio quality is great, and the player interprets gapless playback properly, which is great for live recordings. This new release has added a small cover thumbnail in the playback



Based on cross-platform JavaScript libraries, Museeks in a sleek and minimal music player will run equally well on OS X, Windows, and Linux.

bar, which is one concession to eye candy that we approve of, and we like the quick and easy way you can create playlists. Because it's built atop some JavaScript libraries, you can enable the developer mode to see the code, just as you would with a website, which adds a new dimension to most music players. ■■■

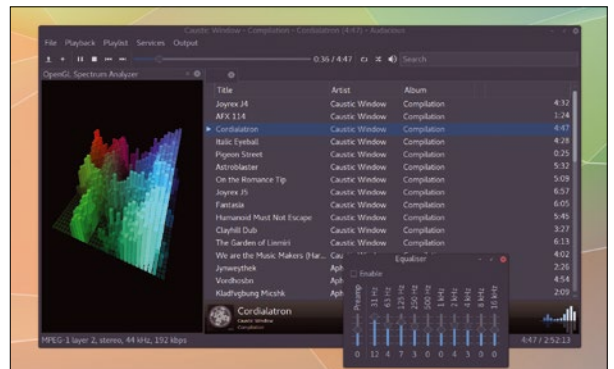
Project Website
<http://museeks.io>

Music Player

Audacious 3.8

If Museeks, above, has piqued your interest in minimal music players, the new release of Audacious is also worth a look. Not quite as fully cross-platform as Museeks (there's a Windows version, too), but without the dependency on bloated JavaScript libraries, its 1.8MB install size will almost fit onto a 3.5-inch floppy disk, if you can remember one of those. This is partly because you can choose between either a Gtk+3 version or a Qt version, not both, and you still need to install a plugin package to generate any sound. This package was a further 6MB on our Arch system, and through these plugins, Audacious is able to talk to PulseAudio, ALSA, SDL and even esoteric outputs like writing audio to disc and Jack.

Audacious is a philosophical descendant of the ancient XMMS, and despite using a modern UI toolkit, it feels very close. The equalizer is a standard addition, for example, and the brilliant OpenGL spectrogram is the only audio visualizer we are tempted to use whilst listening to music. Audacious also plays back everything from MIDI to Nintendo DS chiptunes and can still play audio CDs if you've got an optical drive. It does this with a negligible hit on your system resources and with maximum quality. This latest release works brilliantly on our high-DPI display and lets you do things like enable high bit depths on the output at lower latency or add folders available through GIO-supported protocols (such



Audacious is now more than 10 years old, but that doesn't stop it keeping to its minimal Unix principle of doing one job well.

as FTP). We also really like Robert Cernansky's new plugin for browsing music held on your Raspberry Pi Ampache server. Despite its age, Audacious has kept true to its music player roots, and we find ourselves using it more than we ever have. If you need a quick and easy music player that still has every feature you'll ever need, Audacious is still that player. ■■■

Project Website
<http://audacious-media-player.org>

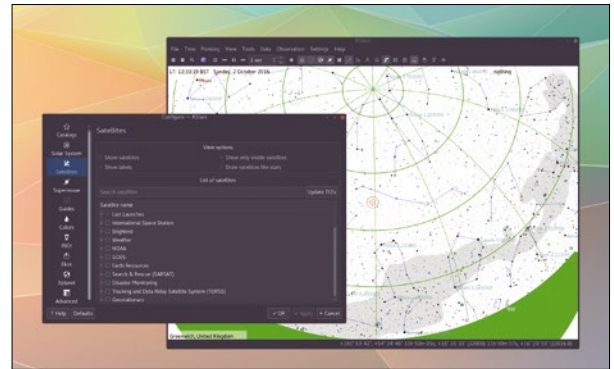
Planetarium

KStars 2.6.0

Stellarium is such a great piece of astronomy software that it's difficult to remember there are several other important projects that take a different look at the stars, and KStars is one of the best. It's been in development for almost as long as KDE has been a desktop, and perhaps because it's bundled as part of KDE's Education metapackages, it doesn't often get the credit it deserves. The project is still being actively worked on, and version 2.6.0 is hopefully an update timely enough for autumnal distribution releases. What makes KStars such a nice tool is that it's good at finding stars and astronomical phenomenon, rather than simulating a view of the night's sky. It still supports up to 100

million stars, 13,000 deep sky objects, and thousands of comets and asteroids, but it won't try and immerse you in a different environment. This means that it doesn't have the same resource requirements, or the same distractions, which makes it better suited to classes, for example, or for running on older hardware.

One of the best things about KStars is that its data sets can be kept up-to-date in ways that other applications can't match. KDE's integrated plugin download actually works, for example, unlike with some other KDE applications, which means you can easily augment KStars with extra catalogs and images. KStars is also brilliant for tracking satellites, and it integrates a TLE (Two Line Elements) download and



Despite calculating the positions of millions of stars, KStars is incredibly quick on modern hardware.

update tool within the Preferences, which is essential for getting and describing the latest, always changing orbital data for objects. It can also track supernovae and download a list of the latest occurrences, if any. Combine this with the classic planetarium interface and sky rendering, and we think KStars is definitely worth another look. ■■■

Project Website
<https://edu.kde.org/kstars>

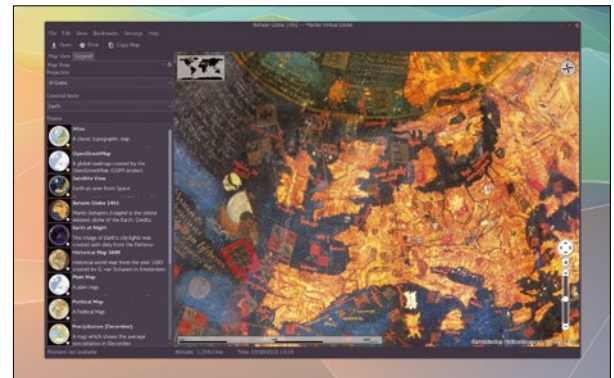
Virtual Globe

Marble 2.0

This KDE application often gets overlooked, as well, and the release of the 2.0 update is the best excuse to look again. Marble is basically an open source version of Google Earth. Unlike Google, all the data used by Marble is open. While you scroll around the 3D globe exploring different parts of the Earth, the data in any layer you're using will be completely "Free," including OpenStreetMap layers, global temperature and precipitation maps (from December and July), weather, real-time clouds, political maps, and even the Behaim Globe from 1492. Plus, you can explore the moon, although the new navigation feature didn't seem to be able to plot a route between Mare Tranquillitatis to Mare Smythii without a detour via Paris. These layers are Mar-

ble's biggest strength, because there's so much to discover while focused on a single location.

Everything is beautifully rendered and presented, even on modest hardware, thanks to the new OpenStreetMap vector rendering, which covers the entire globe and looks fantastic. The satellite view looks good from a distance and serves its purpose perfectly for wide-angle geography, but it can't compete with Google or Microsoft when zoomed in because it's using public domain images, mostly from NASA. We'd love some rich anthropologist to create a similar data set of higher resolution imagery. Marble can be used with a GPS and navigate routes on foot, bike, or road or connect to the flight simulator, Flight Gear, to provide real-time map



Navigate around the Earth like it's 1492.

support. Plugins link to Wikipedia and other local information and, because much of the data can be cached locally, would make an ideal tool for the classroom or travel. The development team are working on an Android version, currently in Beta, that might just bring Marble to a much wider audience, and we can't wait. ■■■

Project Website
<https://marble.kde.org>

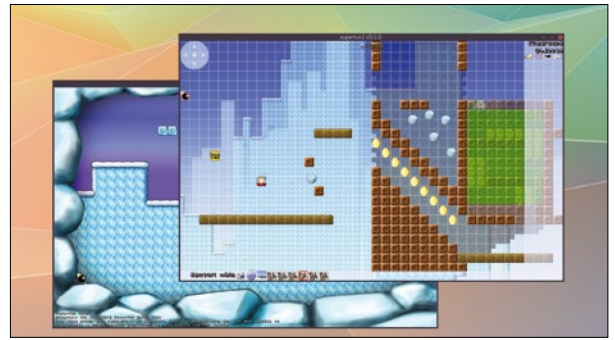
Classic Platformer

SuperTux 0.5.0

There can't be many Linux users who haven't played SuperTux. For many years, before SteamOS seemingly changed everything, it was one of the few genuinely finished and playable games for our platform of choice, although that doesn't include the great work done by Loki, LGP, and Tux Racer. SuperTux is a platform game that's taken a huge dose of inspiration from the various early Super Mario games. You use your lightening-fast reflexes and platforming skills to get your penguin across each level. The thing that sets SuperTux apart is that, because the game has been in development for so long (since 2003), each level is finely tuned, with just the right amount of compulsive addiction. This is

something that doesn't happen overnight, and it's something that SuperTux has over the million other Mario-styled platformers. There is even an online poll for rating all the world 2 levels, so that developers can look at making similar improvements to those levels in a future release.

It's also fantastic that the game is still being actively developed, even if the version number gives no indication of its heritage. This release, for instance, includes an in-game level editor, so you can tweak the levels yourself and see if you can improve on their magic. This is a great way for younger players to not only see how games like this are constructed and programmed, but also to experiment with game mechanics and the idea that what works on



New for this version of SuperTux is a brilliant and game-changing (see what we did there?) level editor.

paper might not always work in practice. The level editor makes this update worth the download on its own, but there are also lots of improvements to some of the levels, the languages, and the game engine. SuperTux can be downloaded for any platform, including installers for other operating systems, and built from the source code, and we highly recommend you grab a copy. ■■■

Project Website
<https://github.com/SuperTux>

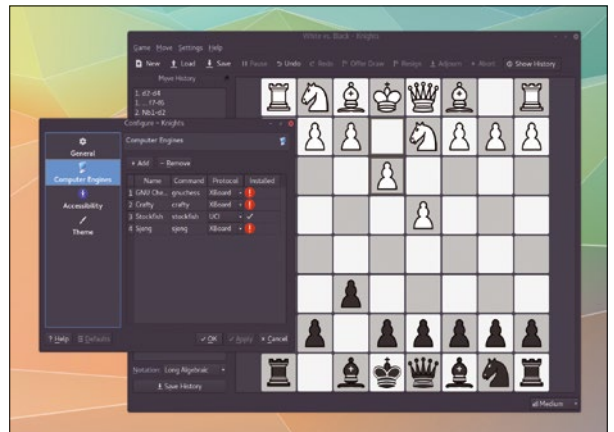
Chess Engine

Stockfish 7

We love playing chess, but it's a difficult game to master. The best way to improve is obviously to play, and computer chess has been a big part of this process. We've been able to play computer-generated chess for decades, from Battle Chess on the Amiga, to Chessmaster on a Gameboy, to DroidFish Chess on Android. DroidFish is brilliant because we love the way it plays chess. Despite our lack of ability, it's always fun and seldom like playing against a computer, which is perhaps the most important thing when it comes to human versus computer outside of Deep Blue versus Garry Kasparov. DroidFish is open source, which is awesome, but so too is the chess engine it uses, called "Stockfish." Chess en-

gines have become interchangeable, with the GUI being like your choice of Linux desktop and the engine being your choice of kernel. Stockfish is rare in the serious world of chess because it plays a brilliant game, and you can play with the source code.

Because Stockfish is only the engine, you can't simply install and run it and expect to play chess. It will install and run, but it's now waiting for a client to communicate with it over the UCI – the universal chess interface. Unfortunately, there aren't that many to choose between for Linux, and many GUIs have been neglected over the last few years. For now, we'd recommend Knights, which detects a running Stockfish and works with other engines, too. From a chess rating perspective,



Stockfish is the chess engine, rather than the GUI, but that means you can play the same game on any desktop of every device.

Stockfish 7 is very good, although we're nowhere near good enough at chess to tell. It certainly wins competitions and is widely respected. We've found that it scales brilliantly, depending on your ability, and plays a very interesting game that isn't the usual train of logic and memory. ■■■

Project Website
<https://stockfishchess.org/>

GAMING ON LINUX

The tastiest brain candy to relax those tired neurons



Michel Loubet-Jambert is our Games Editor. He hasn't had a decent night's sleep since Steam came out on Linux.

The Unity game engine has been making some steady progress, but version 5.5 of the editor is the first where the Linux release has been pushed out under a unified codebase with mainline Unity. This essentially means that shortly after getting the editor ported to Linux, the OS is no longer a second-class citizen, making it a far more enticing option for those looking to develop games natively on this incredibly popular engine. At the same time, although it's most likely a while off, Unity developers have been working on integrating support for the Vulkan API, which is exciting stuff, along with porting to SDL (Simple DirectMedia Layer), which would also be a big milestone once achieved.

In the emulator space, the GameCube and Wii Dolphin emulator can now officially boot every GameCube game – bar one stubborn case. This is a pretty huge milestone for an already impressive emulator, although this doesn't mean all the games run flawlessly yet. Dolphin has already overcome many of the technical limitations of the consoles by being able to increase resolution, add things like anti-aliasing, and play at higher frame rates.

Meanwhile, the DRM-free retailer GOG.com has made changes to its legal agreements that have essentially opened the door for the sale of games with more downloadable content (DLC) and in-game currency. Although these games aren't hugely popular with everyone, it does mean that there is now far greater potential for the roster of games to expand and thus more choice for those who like to avoid DRM.

Rocket League

A go-to multiplayer fix

Web <http://store.steampowered.com/app/252950/>

Price \$19.99/£14.99

Football games have been overdone to the point where yearly releases now consist of roster changes; however, for many, they are still the de facto multiplayer game. Thankfully, Rocket League maintains the universally understood rules that make these games so widespread but changes almost every aspect for those of us bored to death of them.

Aside from the basic premise of two teams, a ball, and goals, the rest is pretty different. On top of cars that can drive up walls and a giant ball, a game of Rocket League does away with all the rules that don't necessarily translate well into video games, such as throw-ins and offsides, resulting in a fast-paced game without interruptions. That said, basketball, ice hockey, and other variants, although fun, are far less commonplace in the available online matches.

Rocket League not only gets cross-platform online multiplayer right on the PC, it also allows Steam users to play against console players, which is a refreshing rarity.



Rocket League tends to be very fast paced and hectic, with all the boring bits cut out.

As always, playing with people you know results in better gameplay given that the typical scenario is otherwise every player going for the ball without any real strategy. It's also worth mentioning that the game has fully functional controller support, which is without a doubt the preferred choice in this case.

The game really hits all the marks when it comes to a multiplayer experience, although it isn't without nitpicks. A more engaging single-player experience or career mode would have been nice, even if it was thrown in at the last second. Even some sort of backstory to the world would be nice. Aside from that, it's hard to find much wrong with the game, and it even does DLC and unlockables right by having no pay-to-win elements or other unfair advantages, while offering customization options to those who want it. Overall, it's a great game to play online – fantastic with friends and somewhat mediocre by yourself. ■■■



Customization options are plentiful and over the top, but offer no in-game advantage.

Master of Orion

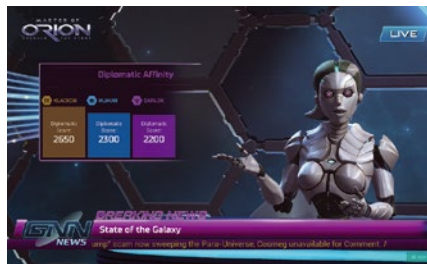
Does it match up to its forerunners?

Web <http://store.steampowered.com/app/298050/>

Price \$29.99/£22.99

The original Master of Orion games are considered among the best of the 4X strategy genre, with a large fan base anticipating the release of their successor. However, since then, a lot of similar games have emerged, so the newest installment has to live up to more than just the hype.

For starters, the combat is fantastic, with excellent close-ups and ships swerving everywhere, getting the look and feel of epic space battles right. The ship customization is decent, while the diplomacy with other races features great animation and an impressive cast of voice actors, many of whom would be familiar to sci-fi fans, giving a certain edge over contemporary competitors like Stellaris.



Master of Orion does well to sell its world and add some light humor.

On the other hand, the game doesn't really push the envelope in any considerable way, and other similar games do many things a bit better. However, it is certainly a huge step up from its predecessors, and it's also worth noting that for those who are dying to re-live the original two games, they are also available on Linux. Overall, the game is still a solid recommend, as long as you're not expecting it to blow away all your expectations for space strategy games. ■■■

Fear Equation

A quirky hidden gem

Web <http://store.steampowered.com/app/428350/>

Price \$14.99/£10.99

It's somewhat rare to see an indie game with decent production values and complex gameplay, so Fear Equation stands out despite being relatively unknown. The game mixes in so many different elements from different genres that it's difficult to summarize concisely, but the main premise is you're the engineer on a train, trying to prevent a nightmarish fog from killing off your passengers.

It's best categorized as a horror game more inclined toward the psychological, with perhaps the most prominent mechanics being rogue-like elements aimed at progressively getting the train to safety; strategic aspects, non-player character (NPC) factions, crafting, and a number of other elements are taken from



Fear Equation is challenging and complicated, but that's part of its charm.

varying genres, which Fear Equation manages to pull off better than some games that have just one of these elements as their exclusive focus. However, this means that a lot of reading and trial and error are needed to come to grips with the game, with a lot of micro-management, although it pays off after the initial overwhelm.

The lack of budget shows at times, but the game does appear to be a cult classic in the making with a fleshed-out world and unique gameplay worthy of having its flaws overlooked. ■■■

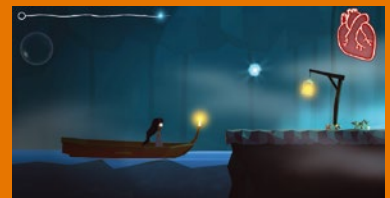
ALSO RELEASED...



What The Box?

This online multiplayer third-person shooter where everyone and everything is boxes makes for an original take on the genre. Unlike other games of this type, with a variety of characters, this version puts the player and identical opponents in maps dominated by identical cardboard boxes, with the best way of not getting killed being to blend in as much as possible. The stealth and silliness provide plenty of fun.

<http://store.steampowered.com/app/527340/>



Selma and the Wisp

This indie puzzle-platformer puts the player in control of a Wisp, which indirectly controls Selma by guiding her through a nightmarish world. It's best to move the Wisp around with a mouse to disarm traps and scare off enemies in clearing a path for Selma and preventing her from meeting an untimely end. The concept is pretty original and makes for some nice casual gameplay.

<http://store.steampowered.com/app/495980/>



RIVE

RIVE is incredibly challenging to the point of frustration, even for a twin stick shooter, but it's also rewarding when you're in sync with the quick reflexes required. RIVE offers a lot of variety for this type of game, incorporating floating and platforming levels, puzzles to vary the pacing, and a story to keep things engaging. If you have a soft spot for classic genres, this is the game for you.

<http://store.steampowered.com/app/278100/>

Ansible Container

Auto Deploy

Combine Ansible and Docker to streamline software deployment

BEN EVERARD

Why do this?

- Quickly create environments that are reproducible on many machines and infrastructures.
- Automate your deployment for ease of reuse.
- Add the word DevOps to your CV and earn more pounds, euros, or dollars.

Computing is bigger than it's ever been. There are now more devices producing and processing more data than last year, and it's only going to get bigger in coming years. For every device in the hands of a user, back-end services are needed to keep everything running smoothly. The tools that have served administrators well for years now struggle to cope with server farms packed to the rafters with computers whirring away and needing maintenance.

Apt and Yum are great at installing software on one machine, but what if you need to update a package on a hundred machines? What if a configuration file needs to be updated but in a slightly different way on hundreds of machines?

Now imagine each of those machines is running a dozen containerized applications that can be launched or stopped depending on how much load is currently on the system. How do you manage now? We'll look at one option for keeping everything humming along smoothly – Ansible Container [1].

Ansible Container uses Docker to create containers that host your code and uses Ansible playbooks to set everything up inside these environments.

The first thing you need is all the software you'll be using to manage the environments you'll create. Ansible Container is written in Python and is available through PIP, but at the time of writing, this was throwing errors on some systems, so I opted to download straight from GitHub.

```
git clone https://github.com/
ansible/ansible-container.git
cd ansible-container
sudo python ./setup.py install
```

The second bit of software you'll need is Docker, which will run behind the scenes and manage the containers themselves. You can run Ansible Container with either Docker Engine (normal Docker) or Docker Machine (which makes it easy to run containers in virtual machines). I opted to go with Docker Machine. Although it may seem a little over-complicated to run containers on separate virtual machines, it eases the set up for running locally.

The install is a little involved and is different on different distros. The basic process will be the same on every distro, but the commands will be different. The following works on Ubuntu 16.04. If you're unsure what to do, the process is well documented for different distros [2].

Docker maintains their own repositories with the latest software versions. With everything changing rapidly, it's best to stay up to date or things may not work, so let's add these repositories to our system. You'll need a couple of things to be able to add the Docker repository to APT. You can get these with:

```
sudo apt-get install
apt-transport-https ca-certificates
```

Cryptographic keys enable APT to verify that the downloads are really coming from Docker. You can install them with:

```
sudo apt-key adv --keyserver
hkp://p80.pool.sks-keyservers.
net:80 --recv-keys
58118E89F3A912897C070AD
BF76221572C52609D
```

Now you're ready to add the Docker repository to APT. Create the file `/etc/apt/sources.list.d/docker.list` with your favorite text editor (you'll need sudo permissions) and add the following line:

```
deb https://apt.dockerproject.
org/repo ubuntu-xenial main
```

With everything set up, you can now grab the latest version of Docker from the repositories:

```
sudo apt-get update
sudo apt-get install docker-engine
```

The Docker service runs in the background controlling the containers that are running. The command-line tool sends instructions to this service about what you want to do, so you need to ensure that the background service is running before you can do anything with the client:

```
sudo service docker start
```

You can make sure that everything is running properly with:

```
sudo docker run hello-world
```

If everything has gone well, you should see a message with the following text:

```
Hello from Docker!
This message shows that your installation
appears to be working correctly.
```

That's Docker Engine set up. Now let's move onto Docker Machine, which is available as a binary file. You can download it ready to run with:

```
sudo curl -L https://github.com/
docker/machine/releases/download/
v0.7.0/docker-machine-`uname
-s`-`uname -m` > /usr/local/bin/
docker-machine
```

You'll need to enable execute permissions before you can run this file. Add this with:

```
sudo chmod +x /usr/local/bin/docker-machine
```

There is one thing left to get, but don't despair, this one's easy to install. You should find Virtual-Box in your distro's repositories. On Ubuntu, you can install it with:

```
sudo apt install virtualbox
```

After all that installing, you're now ready to get into Ansible Container.

First, create a directory for your new container (we'll call ours `ansible-test`), then `cd` into it:

```
mkdir ansible-test
cd ansible-test
```

You'll need a few files for your project, and using the `ansible-container init` command will set everything up for you.

This will create a subdirectory called `ansible` that contains the critical files for your project, and they contain some example code that's commented out. The two most important files are `container.yml` and `main.yml` (Figure 1). Open up `container.yml` and uncomment lines so that it looks like Listing 1.

Listing 1: container.yml

```
01 version: "1"
02 services:
03   web:
04     image: ubuntu:trusty
05     ports:
06       - "80:80"
07     command: ['/usr/bin/dumb-init',
08               '/usr/sbin/apache2ctl', '-D',
09               'FOREGROUND']
10   dev_overrides:
11     environment:
12       - "DEBUG=1"
13 registries: {}
```

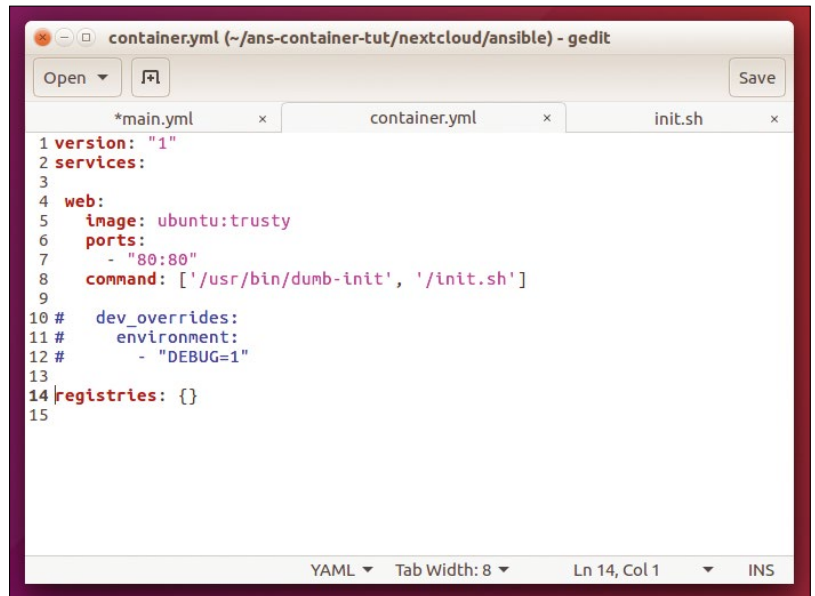


Figure 1: A good text editor will have syntax highlighting for the YML language used in Ansible playbooks.

This file defines the containers you want to run. In this case, it's a single one called `web` that's based on the Ubuntu Trusty image, has port 80 bound to local host port 80, and runs Apache through `dumb-init`.

Now open `main.yml` and uncomment lines until it reads as shown in Listing 2.

This is an Ansible playbook that tells Ansible what to do to configure the container correctly. Let's look at the format of this file. It's text based using the YML markup language. In essence, it's one big list with various items at various depths. A hyphen adds a new item to the list, and adding a two-spaces indent adds another level to the list. In this case, the main list has two items (`hosts: all` and `hosts: web`). These items are plays, and the `hosts` line tells Ansible which hosts to apply this play to. In this project, there is only one host (called `web`), but there can be many. The `hosts: all` line tells Ansible to apply this play to every host that it knows about.

Listing 2: main.yml

```
01 - hosts: all
02   gather_facts: false
03   tasks:
04 - raw: which python || apt-get update
05 - raw: (which python && which aptitude) ||
06   apt-get install -y python python-app
07   aptitude
08 - hosts: web
09   tasks:
10     - name: Upgrade APT
11       apt: upgrade=yes
12     - name: Install ca-certificates
13       apt: name=ca-certificates state=latest
14     - name: Install dumb-init
15       apt: deb=https://github.com/Yelp/
16         dumb-init/releases/download/v1.0.2/
17         dumb-init_1.0.2_amd64.deb
18     - name: Install Apache
19       apt: name=apache2 state=latest
```

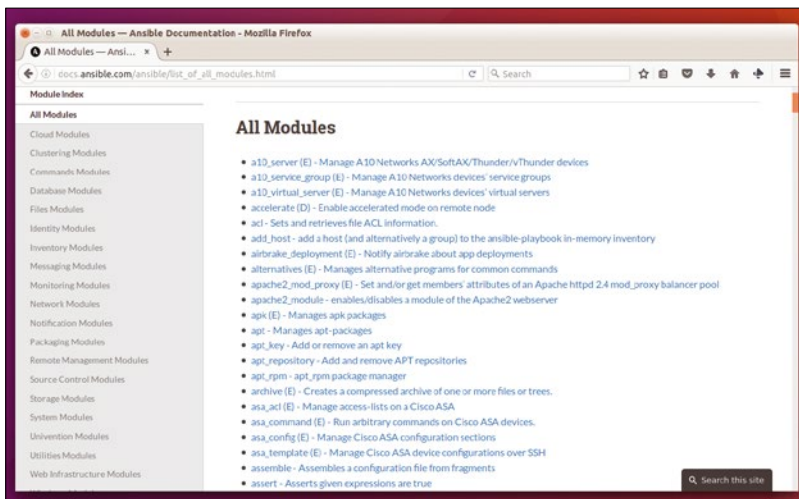


Figure 2: There are far more Ansible modules than we can cover here, but you can find the documentation on the Ansible website.

Inside the play, there’s a sublist called tasks, which is the sequence of things that you want Ansible to do. Tasks typically have a name and an action (the name is optional but useful). The action has to link to an Ansible module that defines what you want to do at each stage. In the play-book in Listing 2, there are two different types of action: `raw`, which is just a command to be run via SSH, and `apt`, which uses APT to install or manipulate packages. There are many modules that enable us to do other things (Figure 2).

You can now build the container by moving back to the directory in which you ran the first `ansible-container` command and running:

```
ansible-container build
```

This may take a little while to run, but it should end by successfully building the machine. There will be some terminal output like:

```
Exporting built containers as images...
Committing image...
Exported web with image ID
```

Once this has completed, you can start the container with:

Listing 3: More main.yml Config

```
01 - hosts: all
02   gather_facts: false
03   tasks:
04     - raw: which python || apt-get update
05     - raw: (which python && which aptitude) ||
      apt-get install -y python python-apt aptitude
06 - hosts: web
07   tasks:
08     - name: Install dumb init
09       get_url: dest=/usr/bin/dumb-init
      url=https://github.com/Yelp/dumb-init/releases/
      download/v1.0.2/dumb-init_1.0.2_amd64 mode=0775
      validate_certs=no
```

```
ansible-container run
```

This will spin up a new Docker container and bind it to port 80 so you can access the web server. Point your web browser to `http://localhost` and you should see an Apache page.

Up to the Clouds

Now we’ll move on to a more complex example – installing a Nextcloud server. This will have exactly the same structure as before (with a play-book in `main.yml` that tells Ansible how to configure your machine), but there will be more to do to get the server up and running.

To begin, create a new directory and run `ansible-container init` to set up the files needed. Again, the majority of the configuration will be in the `main.yml` file. This starts in the same way as in the previous example (Listing 3).

You’ve got `dumb-init` installed now, which allows you to run a single command as you start the machine. However, your container will need to run two services (Apache and MySQL), so you’ll need to create a script to run the init process to start both of these (see Listing 4).

Two modules are used to create the `init.sh` file and `lineinfile`, which you use to add the lines you need. `lineinfile` is far more powerful than this; it can be used to search and replace particular lines by regular expressions. However, you don’t need this capability. You might have noticed that the `init.sh` file is placed in the root and anyone can execute it.

We’re side-stepping some issues by setting it up this way. To deploy this service properly, you wouldn’t want to set it up like this, but then you probably wouldn’t want to run the database in the container as well (at least not without some clustering setup to allow it to repopulate itself if the container is recreated). This setup is designed to test out software before deploying it to a permanent home.

The set of tasks is to grab all the packages that you’ll need (Listing 5).

Here you can see how to iterate through a list of items with each option being inserted as `{{ item }}` in the command.

Listing 4: Run and Start Apache and MySQL

```
01 - name: create init file
02   file: path=/init.sh state=touch mode=a+rx
03 - name: copy over init file1
04   lineinfile: dest=/init.sh line="#!/usr/bin/dumb-init /bin/sh"
05 - name: copy over init file2
06   lineinfile: dest=/init.sh
      line="/usr/sbin/apache2ctl -D FOREGROUND &"
07 - name: copy over init file3
08   lineinfile: dest=/init.sh
      line="/usr/bin/mysqld_safe"
```


Your Roadmap to the Open Hardware Revolution ...

An exciting world of projects, tips, and skill-building tutorials awaits you in every issue of Raspberry Pi Geek.

Order your subscription today and tune in to the revolution!

shop.linuxnewmedia.com

Print Sub

Carry our easy-to-read print edition in your briefcase or backpack - or keep it around the lab as a permanent reference!

Digital Sub

Our PDF edition is a convenient option for mobile readers.

6 print issues with 6 DVDs or
6 digital issues for only

\$59.95 £37.50 €44.90



We're also in
Google Play
and the
iTunes Store!



iTunes Store



Google US



Google UK

Ansible Roles and Galaxy

In this tutorial, we've built a playbook that sets up our server. It works, and we can use it. However, it's not the most reusable script. For example, if you now wanted to set up another server that used MySQL and Apache, you'd have to copy or rewrite the part of the playbook that installs and enables this. In this example, everything's quite simple, so it wouldn't be much work to do this. In more involved installs, however, this becomes a lot of duplication of effort and code. Ansible has a couple of ways of reducing the duplication that can make it much easier to manage large numbers of different configurations.

- **Tasklists.** The simplest option for sharing work between projects is tasklists. These are just YML files that contain tasks, and you can include them in playbooks with a line like: `include: <tasklist-file>`.
- **Roles.** Rather than think about the tasks you want to perform on a server, we can think of

the roles you want the server to perform. For example, you might want a server to perform the role of a LAMP web server. Ansible allows you to encapsulate the tasks necessary for this to work as a *Role*. These roles can then be added to hosts, and Ansible will work out what needs to happen to the server to enable it to perform that role.

- **Galaxy.** Well-defined roles are often not project-specific. For example, the LAMP web server will just need a few variable changes to make it applicable to almost any project that needs a server like this. Ansible Galaxy [3] is a website that enables Ansible users to share roles so that they can be easily added to different projects without everyone having to create them from scratch (Figure 4). If you need to set up a piece of open source software, there's a good chance that someone has already shared a role for this.

With everything set up, you only have to download the latest version of Nextcloud (Listing 6).

This uses the `unarchive` command to pull the software from a remote repository straight into the web root on the host. The `copy=no` option to `unarchive` is needed because of a bug in Ansible when using `remote_src` on `unarchive`. The final task here makes sure that files in the webroot are owned by the correct user.

With `main.yml` set up, you just need to adjust `container.yml`. The only difference between this and the previous example is that you need to tell

`dumb-init` to run the `init.sh` script rather than the Apache start script:

```
command: ['/usr/bin/dumb-init', '/init.sh']
```

Now you're ready to launch our containers with the same commands as before:

```
ansible-container build
ansible-container run
```

If you point your web browser to `localhost/nextcloud`, you'll be able to enter the details needed to finish the Nextcloud install. The database login is root with no password. Again, this isn't the most secure setup, but again, this database setup isn't designed for live use. The best setup depends a lot on your current system, whether you've already got a database server you can use, and the amount of load you're expecting on the system.

When the containers are running, you might want to interact with them – maybe to just have a poke around in the system or maybe to

Listing 5: Grab the Packages

```
01 - name: Upgrade APT
02   apt: upgrade=yes
03
04 - name: Install Everything
05   apt: name={{item}} state=installed
06   with_items:
07     - apache2
08     - git
09     - mysql-server
10     - unzip
11     - libapache2-mod-php5
12     - php5-gd
13     - php5-json
14     - php5-mysql
15     - php5-curl
16     - php5-intl
17     - php5-mcrypt
18     - php5-imagick
```

Listing 6: Download Nextcloud

```
01 - name: download nextcloud
02   unarchive: copy=no remote_src=yes
              src=https://download.nextcloud.com/
              server/releases/nextcloud-10.0.0.zip
              dest=/var/www/html/
03
04 - name: set permissions
05   file: path=/var/www/htmlrecurse=yes
         owner=www-datastate=directory
```

Why Not Dockerfiles?

Docker does of course have its own provisioning system – Dockerfiles – that can be used to set up containers. However, the problem with Dockerfiles is that they can only be used with Docker. Effectively, they bundle everything up into a single technology stack for both your provisioning and hosting.

With Ansible Container, you separate this out. Docker is used to run the images, but the provisioning is done via Ansible playbooks. This means you can use Ansible Container to create your development environment. You can also deploy with Ansible Container if you wish; but, if this isn't the right choice for you, the same playbooks you used to build your containers can be used to set up virtual servers or physical servers.

By decoupling the provisioning from the hosting, Ansible Containers give you far more flexibility than Dockerfiles.

add or remove something without having to rebuild the container. Ansible Container creates your containers, but they're run through Docker, so this is the tool you'll need to access them (See the box entitled "Why Not Dockerfiles?"). You can see a list of currently running Docker containers with:

```
sudo docker ps
```

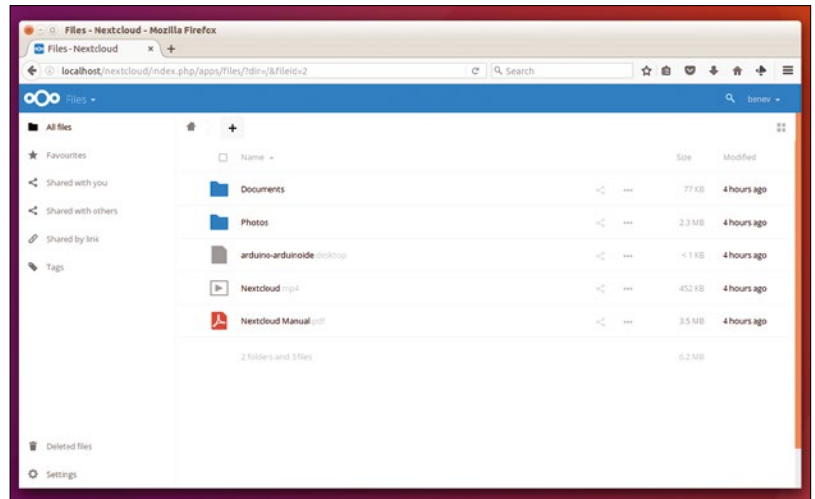


Figure 3: Our Nextcloud server running in a Docker container provisioned by Ansible.

You can then attach a Bash session (as root) to the container using the ID listed in the first column of the output of the previous command with:

```
sudo docker exec -i -t <id> /bin/bash
```

With this, you have a container up and running Nextcloud, and you can connect to it via a terminal to take care of any admin (Figure 3). By using a combination of Ansible and Docker, you can easily customize your containers to perform almost any function and share the configurations online using Ansible Galaxy (see boxout). You can then deploy your handiwork on almost any infrastructure, giving you a great combination of ease, flexibility, and power. ■■■

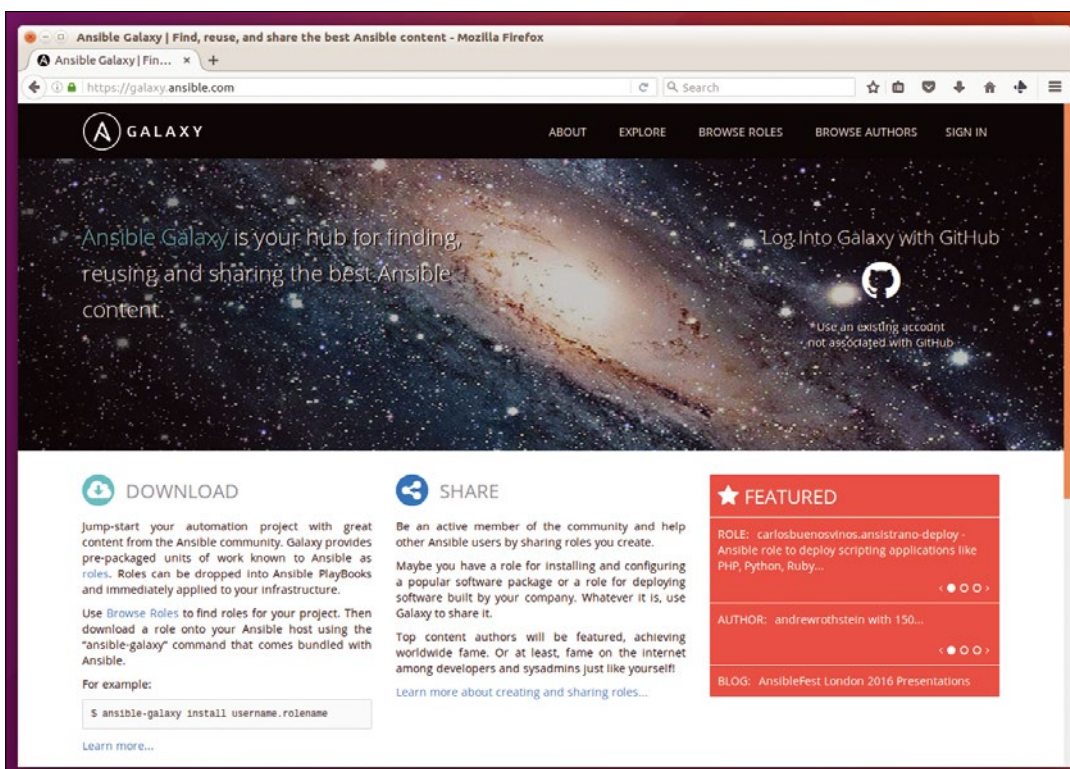


Figure 4: Set up servers automatically by downloading Ansible roles from the Galaxy website.

Info

- [1] Ansible Container: <https://www.ansible.com/ansible-container>
- [2] Docker Machine: <https://docs.docker.com/machine/install-machine/>
- [3] Galaxy: <https://galaxy.ansible.com>

Multi-Tasking

Turbo-charge your command-line workflow with Tmux

MIKE SAUNDERS

Why do this?

- Run multiple CLI apps in the same window
- Save your SSH sessions for logging in later
- Create custom layouts for your workflow

Unless you're some kind of hard-core raw X Window System user who doesn't need to resize or move around any of your applications, chances are that you're running a window manager of some sort. This may be a stand-alone window manager like Fluxbox, a tiling variant like i3, or perhaps one that's part of a larger desktop environment, such as Xfce, KDE, or Gnome. Window managers are one of the most essential tools for our work – but can they actually be used in text mode?

Well, yes. It might sound a bit weird at first, given that the command line is all about typing and viewing text and doesn't have the same requirements as a pointy-clicky, frill-laden desktop environment. But consider the typical work you do at a Bash prompt: Do you have a single terminal running at maximum screen resolution? Unless you're still rocking a netbook, chances are you have multiple terminals in action at any one point, using your window manager to organize them.

Now, imagine you could leave your window manager out of this and do the work of organizing different sessions inside a single terminal

window. This is what Tmux, the “terminal multiplexer” does – and it does it very well [1]. With Tmux, you effectively have a text-mode window manager available at the command line, so you can create different views (e.g., with a big main command-line view for your day-to-day work and smaller views next to it for monitoring logs or IRC channels).

There are several benefits to doing this inside Tmux rather than your window manager. For starters, you can use Tmux when SSH'd into a remote machine and disconnect from that remote machine at any point, leaving all the sessions running. Once you have everything set up nicely, even if your connection drops, everything will be back where you left it once you reconnect. Compare this to having several terminal windows open on your desktop and needing to log in and start processes again in each one.

Plus, Tmux has minimal system requirements and can be installed almost everywhere. If you move around a lot between computers, Linux distros, or even operating systems, you can rely on Tmux to look, feel, and work the same. Oh, and it looks rather funky and scores you geek points as

Building Tmux 2.3 from Source Code

A new version of Tmux was released in September 2016, so we couldn't resist installing it in time to write this feature. If the new version isn't in your distro's package repositories yet, or you just like to build software from its source code, then here's a quick guide.

First, grab the latest release [1] and save it to your home directory. Then open a terminal and enter:

```
tar xfv tmux-2.3.tar.gz
cd tmux-2.3
```

This extracts the compressed source-code archive and switches into the resulting directory. Next, run:

```
./configure
```

This sets up the build environment and checks that you have the necessary dependencies installed. Tmux is fairly light on dependencies, so you won't have to hunt down lots of obscure stuff to build it. The two main packages you need are the development headers for Ncurses and Libevent, so find those in your package manager. When `./configure` has run successfully, enter:

```
make
sudo make install
```

This will build Tmux and install it (by default into `/usr/local`). Then you can run it by entering `tmux` as in the main text.

well. So, over the next few pages, I'll show you how to master it and boost your command-line productivity.

Getting Set Up

Tmux is available in almost every Linux distro under the sun, so you should be able to grab it via your package manager. Indeed, some distros include it in the stock installation. If, for some reason it's not available for your distro – or you want to get the very latest version – then see the “Building Tmux 2.3 from Source Code” box. Once you have it installed, start it in a terminal window by entering:

```
tmux
```

The screen will clear, and you'll be left at a new command prompt. Doesn't look like much, right? Switch your attention to the bottom of the terminal, and you'll notice a long green bar spanning the whole width of the window (Figure 1). This is known as the “status line” in Tmux parlance and is highly configurable; even in a default, stock Tmux setup, it still contains useful information, though.

For example, on the left-hand side you'll see a list of running programs – a bit like a traditional desktop taskbar. In this case, it's only showing Bash as that's the only program running. In a moment, however, we'll fill up this section. On the right-hand side of the status line, you'll see the hostname of the machine you're using, a clock, and the current date.

So, let's work with Tmux's most prominent feature: the ability to run multiple programs side by side. Enter `nano` to start a text editor, and then press `Ctrl+B` followed by `C` on its own (both lower-case). Eek – Nano has disappeared! You're presented with a new command line. What has happened here?

Well, `Ctrl+B` is the key combo used to send a command to Tmux. In most of your interaction with Tmux, you'll use this to get the program's attention – and then press another key to perform a specific action. In this case, we hit `Ctrl+B` to alert Tmux, and then `C`, which means “create a new window” (Figure 2). If you look at the left-hand side of the status line, it should say something like this:

```
[0] 0:nano- 1:bash*
```

The number in square brackets refers to the Tmux session in use – you can ignore that for now. Then there's a list of windows, with associated numbers, and the window we're currently viewing has an asterisk (*) next to it. So, you can see that the Nano we started is cur-

```
TMUX(1) BSD General Commands Manual TMUX(1)
NAME
  tmux - terminal multiplexer
SYNOPSIS
  tmux [-2CluvV] [-c shell-command] [-f file] [-L socket-name]
      [-S socket-path] [command [flags]]
DESCRIPTION
  tmux is a terminal multiplexer: it enables a number of terminals to be
  created, accessed, and controlled from a single screen. tmux may be
  detached from a screen and continue running in the background, then later
  reattached.

  When tmux is started it creates a new session with a single window and
  displays it on screen. A status line at the bottom of the screen shows
  information on the current session and is used to enter interactive com-
  mands.

  A session is a single collection of pseudo terminals under the management
  of tmux. Each session has one or more windows linked to it. A window
  occupies the entire screen and may be split into rectangular panes, each
  Manual page tmux(1) line 1 (press h for help or q to quit)
[0] 0:man+ "raspberrypi" 08:09 04-Oct-16
```

Figure 1: Here's what Tmux looks like out of the box, with the green status line at the bottom.

rently sitting idle in window number 0, whereas the Bash prompt you're viewing right now is in window 2.

Switching between windows is easy: Just press `Ctrl+B` followed by the number. To switch back to Nano, you'd press `Ctrl+B` followed by `0`. You can keep opening new windows using the aforementioned `Ctrl+B` and then `C` combo; to close them, just log out of the Bash session (e.g., with `exit` or `Ctrl+D`).

Now, when you have a lot of windows open, you may be doing a job that requires switching between just two of them for a while (e.g., editing a configuration file while reading a manual page). Having to use the specific window numbers each time can become rather tedious in this case, but Tmux has a solution: `Ctrl+B` followed by `L`. This switches to the “last viewed” window – a bit like hitting `Alt+Tab` once in a traditional desktop environment.

This is the killer feature of Tmux: the ability to run multiple programs in the same terminal

Figure 2: When you create a new window, you can see it on the left of the status bar – in this case, the “1:bash*” part.

```
mike@raspberrypi:~$ ls -l /
total 84K
drwxr-xr-x  2 root root 4.0K May 12 09:50 bin
drwxr-xr-x  5 root root 16K Jan  1 1970 boot
drwxr-xr-x 14 root root 3.2K Oct  2 17:26 dev
drwxr-xr-x 90 root root 4.0K Oct  2 13:51 etc
drwxr-xr-x  3 root root 4.0K May 12 09:42 home
drwxr-xr-x 17 root root 4.0K May 12 09:48 lib
drwx----- 2 root root 16K Mar 18 2016 lost+found
drwxr-xr-x  2 root root 4.0K Mar 18 2016 media
drwxr-xr-x  2 root root 4.0K Mar 18 2016 mnt
drwxr-xr-x  3 root root 4.0K Mar 18 2016 opt
dr-xr-xr-x 81 root root  0 Jan  1 1970 proc
drwx----- 3 root root 4.0K Oct  2 13:52 root
drwxr-xr-x 20 root root 660 Oct  2 17:27 run
drwxr-xr-x  2 root root 4.0K May 12 09:51/sbin
drwxr-xr-x  2 root root 4.0K Mar 18 2016/srv
dr-xr-xr-x 11 root root  0 Jan  1 1970/sys
drwxrwxrwt  8 root root 4.0K Oct  4 08:09/tmp
drwxr-xr-x 10 root root 4.0K Mar 18 2016/usr
drwxr-xr-x 12 root root 4.0K May 12 12:19/var
mike@raspberrypi:~$
[0] 0:man- 1:bash* "raspberrypi" 08:09 04-Oct-16
```

window. But, it becomes especially useful in terms of its ability to keep sessions running, even when you close the terminal. To try this out, in your Tmux session, press Ctrl+B followed by D to “detach” from Tmux and return to the prompt. You will see something like:

```
[detached (from session 0)]
```

What’s important here is that Tmux is still running, and the programs it started are still running as well. You can close the terminal window, open a new one, and then enter the following to reattach to the Tmux you started before:

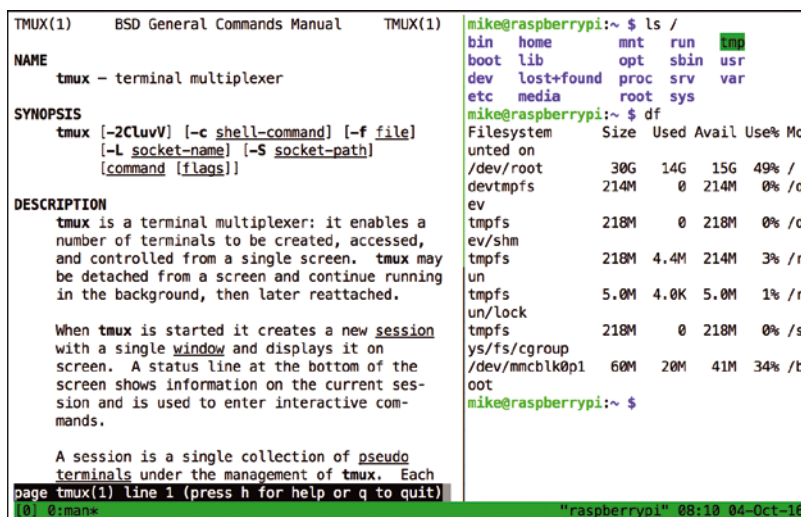
```
tmux a
```

How’s that? It’s an extremely useful feature that really comes to life when you’re SSH’ing into other machines. If you do a lot of admin work on a server, for instance, or a lot of playing around via SSH on a Raspberry Pi, you can log in, run Tmux, and set up the session to your liking – with different programs in different windows. If you need to disconnect for a while, just detach and close the terminal window. Later, you can SSH back in, run `tmux a` to reconnect, and everything was exactly how you left it. Indeed, you’ll wonder how you managed to live without Tmux before this point (well, unless you were using GNU Screen, which is similar but not quite as awesome).

Custom layouts

So far we’ve been working with Tmux windows that fill the whole terminal window – but it’s possible to split up the screen into multiple sections called “panes” as well. Create a new Tmux window with Ctrl+B followed by C, and then use Ctrl+B with percentage sign (%) to split the window down the middle (vertically) into two Bash sessions (Figure 3).

Figure 3: Here we used Ctrl+B followed by % to create a split down the middle of the screen, for two panes.



Now you have left and right panes, and you can switch between them by using Ctrl+B followed by the arrow keys. To identify the pane you’re currently using, just see where the cursor is. To close a pane, you can simply exit the Bash prompt or use Ctrl+B then X to kill it.

Now, you probably don’t want a direct split down the middle – it’s more likely that you want a big pane on the left for doing your work, and a smaller one on the right for monitoring things. Fortunately, Tmux lets you resize panes to arbitrary sizes, very much like a normal window manager. To do this, use the usual Ctrl+B prefix, followed by Ctrl+Left/Right (the arrow keys). This changes the size in one-character increments – so it may take a few repeated presses to get the setup that you want. (In some setups, using Ctrl+B followed by Alt+Left/Right will resize in steps of five characters.)

But what about horizontal splits? Those are possible, too, using Ctrl+B followed by double quotation marks ("). This results in a window with a pane at the top and another at the bottom, switchable and resizable using the arrow key commands mentioned before (but with Left/Right replaced by Up/Down).

Note that it’s possible to combine horizontal and vertical splits – so you could have one big pane on the left occupying around 70 percent of the window space, doing some editing work, while on the right you have two small panes, stacked on top of each other, with Htop and an IRC client running, for example. Tmux is tremendously flexible and caters to all kinds of layouts, and with time you’ll be constructing elaborate setups with multiple sessions containing multiple windows containing multiple panes. If you spend a lot of time at the CLI, it’s bliss.

Colors and Theming

At this point, you may be slowly getting bored with Tmux’s default garish green in the status line – but as with almost everything in the program, it’s highly configurable. You can change colors on the fly, but a better way to make them permanent is to create a `.tmux.conf` file in your home directory. The commands you add in there are mostly human-readable, for example:

```
set -g status-bg red
set -g status-fg white
```

As you might have guessed, this changes the status bar colors so that the background is red and the foreground is white (Figure 4). Note that you can also specify RGB hexadecimal values, like so:

```
set -g status-fg "#0000ee"
```

But, of course, your terminal will need to support this – many only offer a limited range of colors. You'll often see messages on the status line in Tmux, especially if you try to do something that's not possible (e.g., moving to a window that doesn't exist). It's also possible to configure the colors of these messages like so:

```
set -g message-bg yellow
set -g message-fg black
```

Along with the status line, the borders separating panes can be configured to use different colors as well:

```
set -g pane-border-fg blue
set -g pane-active-border-fg red
```

What about the position of the status line? If you'd prefer to have it at the top of the screen, that's also easy to do:

```
set -g status-position top
```

Once you have your settings neatly organized in `.tmux.conf`, you can restart Tmux to use them. Alternatively, you can tell Tmux to re-read the config file by following the instructions in the "Changing the Command Key" box. For more options for customizing the colors and status bar layout, check out the handy online guide [2].

Another good source of documentation (albeit rather dry and less hands-on) is the manual page, which you can access with `man tmux` or read online [3]. This provides a detailed list of all the keybindings available, along with commands that can be executed inside Tmux or placed into the `.tmux.conf` file.

You can even enable mouse support, allowing you to manipulate Tmux's panes with the rodent, making it a real hybrid between a text-mode and desktop window manager. But we doubt many people will want to use that. As with Vim, the real beauty of Tmux is found on the keyboard and in the fact you can do almost anything without having to move your fingers very far from the home row. ■■■

The Author

Mike Saunders thinks these newfangled "graphical user interfaces" will never really take off. Get off his lawn.

Info

- [1] Tmux: <https://tmux.github.io/>
- [2] Online guide: <http://tinyurl.com/jbd2ygv>
- [3] Man pages: <http://man.openbsd.org/OpenBSD-current/man1/tmux.1>

The screenshot shows a terminal window with a tmux status bar at the bottom. The status bar includes window titles like 'K10halt [R0]' and 'K10halt [readonly] 83L, 1336C', and pane information like 'Top .tmux.conf 5,22 ALL'. The terminal content shows the output of 'ls /' and 'uptime' commands, and a snippet of the /etc/passwd file. The tmux configuration file (.tmux.conf) is visible in the background, showing settings for message colors and status position.

Figure 4: Here we're taking it to the max, with multiple splits and a custom red status bar.

Changing the Command Key

If you're an Emacs user, you probably don't mind tapping `Ctrl+B` regularly to activate Tmux commands. But you may also have a single key sitting around doing very little on your keyboard, and you'd rather use that instead. This is possible in Tmux and requires a few commands in `.tmux.conf` in your home directory.

For example, we find that we very rarely use the back tick (``) key on our keyboards, but it's close to hand, so we want to use this single keypress instead of the `Ctrl+B` combination. In our `.tmux.conf`, we have the following:

```
unbind C-b
set-option -g prefix `
bind ` send-prefix
```

What if you really need to use a back tick now – for instance, when creating a multi-part command in Bash? The answer is simple: Just press back tick twice. This tells Tmux to ignore the first one and interpret the second as standard keyboard input.

Now you can create a new window with `` followed by C, or switch between the two most recent windows with `` followed by L, and so forth. Note that you can change your `.tmux.conf` and tell a running Tmux session to update itself accordingly by pressing `Ctrl+B` followed by colon (:), to open up a command line in the status line, somewhat like in Vi(m). Then enter:

```
source .tmux.conf
```

This tells Tmux to re-read the config file and change its settings.

FEATURED EVENTS

Users, developers, and vendors meet at Linux events around the world. We at *Linux Magazine* are proud to sponsor the Featured Events shown here. For other events near you, check our extensive events calendar online at <http://linux-magazine.com/events>.

If you know of another Linux event you would like us to add to our calendar please send a message with all the details to events@linux-magazine.com



Node.js Interactive

Date: November 29–December 2, 2016

Location: Austin, Texas

Website: <http://events.linuxfoundation.org/events/node-interactive>

Node.js Interactive is the marquee event for JavaScript developers, companies that rely on Node.js, and vendors that support them with tools, training, and other services. Join the fun with Breakout, Code & Learn, and Un-conference sessions, as well as the usual booth crawl.

IT-Tage 2016

Date: December 12–15, 2016

Location: Frankfurt, Germany

Website: <https://www.ittage.informatik-aktuell.de>

The “IT Days” conference offers up to six parallel tracks with more than 160 lectures and sessions on a wide range of developer, management, and admin topics. Keynote speakers include Michael “Monty” Widenius, of MySQL and MariaDB, Berni Schiefer, IBM Fellow, IBM Analytics, and others.

DevConf Panamá 2017

Date: February 1–3, 2017

Location: Panama City, Panama

Website: <http://devconfpanama.com/#/>

DevConf appears for the first time in Panama, with a focus on trends in software development and technology, featuring popular frameworks such as Drupal 8, Symfony 2, AngularJS, Node.js, and Backdrop CMS. Connect with and learn from the people behind successful software projects.

EVENTS

Web Summit	November 7–10	Lisbon, Portugal	https://websummit.net
ApacheCon Europe	November 16–18	Seville, Spain	http://events.linuxfoundation.org/events/apachecon-europe
UbuCon Europe	November 18–20	Essen, Germany	http://ubucon.org/de/events/ubucon-europe/
LinuxDay in Vorarlberg	November 26	Dornbirn, Austria	http://www.linuxday.at
AnDevCon	November 29– December 1	San Francisco, CA	http://www.andevcon.com-
Node.js Interactive North America	November 29– December 2	Austin, Texas	http://events.linuxfoundation.org/events/node-interactive
Open Source Monitoring Conference	November 29– December 2	Nuremberg, Germany	https://www.netways.de/events_schulungen/osmc/uebersicht/
SPTechCon	December 5–8	San Francisco, California	http://www.sptechcon.com/venue
IT-Tage 2016	December 12–15	Frankfurt, Germany	https://www.ittage.informatik-aktuell.de
DevConf Panamá 2017	February 1–3	Panama City, Panama	http://devconfpanama.com/#/

CONTACT INFO

Editor in Chief
Joe Casad, jcasad@linux-magazine.com

Managing Editor
Rita L Sooby, rsooby@linux-magazine.com

Localization & Translation
Ian Travis

News Editor
Swapnil Bhartiya

Copy Editors
Amber Ankerholz, Amy Pettle

Layout
Dena Friesen, Lori White

Cover Design
Dena Friesen, Lori White

Cover Image
© Roberto Scandola, 123RF.com; Greg Kroah
Hartman image © Sebastian Oliva CC-BY-SA-3.0

Advertising – North America
Ann Jesse, ajesse@linuxnewmedia.com
phone +1 785 841 8834

Advertising – Europe
Brian Osborn, bosborn@linuxnewmedia.com
phone +49 89 99 34 11 48

Publisher
Brian Osborn, bosborn@linuxnewmedia.com

Marketing Communications
Gwen Clark, clark@linuxnewmedia.com
Linux New Media USA, LLC
616 Kentucky St.
Lawrence, KS 66044 USA

Customer Service / Subscription
For USA and Canada:
Email: cs@linuxpromagazine.com
Phone: 1-866-247-2802
(Toll Free from the US and Canada)
Fax: 1-785-856-3084
For all other countries:
Email: subs@linux-magazine.com

Phone: +49 89 99 34 11 67
Fax: +49 89 99 34 11 98

www.linuxpromagazine.com – North America
www.linux-magazine.com – Worldwide

While every care has been taken in the content of the magazine, the publishers cannot be held responsible for the accuracy of the information contained within it or any consequences arising from the use of it. The use of the disc provided with the magazine or any material provided on it is at your own risk.

Copyright and Trademarks © 2016 Linux New Media USA, LLC.

No material may be reproduced in any form whatsoever in whole or in part without the written permission of the publishers. It is assumed that all correspondence sent, for example, letters, email, faxes, photographs, articles, drawings, are supplied for publication or license to third parties on a non-exclusive worldwide basis by Linux New Media USA, LLC, unless otherwise stated in writing.

Linux is a trademark of Linus Torvalds.

All brand or product names are trademarks of their respective owners. Contact us if we haven't credited your copyright; we will always correct any oversight.

Printed in Germany

Distributed by COMAG Specialist, Tavistock Road, West Drayton, Middlesex, UB7 7QE, United Kingdom

LINUX PRO MAGAZINE (ISSN 1752-9050) is published monthly by Linux New Media USA, LLC, 616 Kentucky St., Lawrence, KS, 66044, USA. Periodicals Postage paid at Lawrence, KS and additional mailing offices. Ride-Along Enclosed. POSTMASTER: Please send address changes to Linux Pro Magazine, 616 Kentucky St., Lawrence, KS 66044, USA.

Published monthly in Europe as Linux Magazine (ISSN 1471-5678) by: Sparkhaus Media GmbH, Putzbrunner Str. 71, 81739 Munich, Germany.



AUTHORS

Erik Bärwaldt	24, 36
Swapnil Bhartiya	8, 14
Zack Brown	11
Bruce Byfield	54
Joe Casad	3, 63
Nate Drake	44
Ben Everard	72, 86
Markus Feilner	32
Andrew Gregory	69
Karsten Günther	18
Heike Jurzik	22
Kristian Kißling	32
Klaus Knopper	48
Charly Kühnast	62
Martin Loschwitz	40
Michel Loubet-Jambert	84
Sebastian Mogilowski	32
Graham Morrison	78
Simon Phipps	64
Dmitri Popov	58
Mike Saunders	66, 92
Tim Schürmann	50
Valentine Sinitsyn	74

United States Postal Service Statement of Ownership, Management, and Circulation

1. Publication Title: Linux Pro Magazine		
2. Publication No.: 1752-9050		
3. Filing Date: 09/30/16		
4. Issue Frequency: Monthly		
5. No. of Issues Published Annually: 12		
6. Annual Subscription Price: \$99.95		
7. Complete Mailing Address of Known Office of Publication: 616 Kentucky, Lawrence, KS 66044;		
Contact Person: Brian Osborn; Telephone: 785-856-3080		
8. Complete Mailing Address of Headquarters or General Business Office of Publisher: 616 Kentucky, Lawrence, KS 66044		
9. Full Names and Complete Mailing Addresses of Publisher, Editor, and Managing Editor: Brian Osborn, Publisher, 616 Kentucky, Lawrence, KS 66044; Joe Casad, Editor, 616 Kentucky, Lawrence, KS 66044; Rita Sooby, Managing Editor, 616 Kentucky, Lawrence, KS 66044		
10. Owner: Linux New Media USA, LLC, 616 Kentucky, Lawrence, KS 66044; Stockholders: Sparkhaus Media GmbH, Putzbrunner Str. 71, 81739 Munich, Germany		
11. Known Bondholders, Mortgages, and Other Security Holders Owning or Holding 1 Percent or More of Total Amount of Bonds, Mortgages, or Other Securities: None		
12. Tax Status: Has not changed during preceding 12 months		
13. Publication Title: Linux Pro Magazine		
14. Issue Date for Circulation Data: October 2016		
I certify that all information furnished on this form is true and complete. Brian Osborn, Publisher		
15. Extent and Nature of Circulation		
a. Total Number of Copies (Net Press Run)	7314	6531
b. (1) Paid Outside-County Mail Subscriptions	1045	1013
b. (2) Paid In-County Subscriptions	0	0
b. (3) Sales Through Dealers & Carriers, Street Vendors, Counter Sales	2850	2542
b. (4) Other Classes Mailed Through the USPS	175	123
c. Total Paid Distribution	4070	3678
d. (1) Outside-County	0	0
d. (2) In-County	0	0
d. (3) Other Classes Mailed Through the USPS	0	0
d. (4) Free Distribution Outside the Mail	375	200
e. Total Free Distribution	375	200
f. Total Distribution	4445	3878
g. Copies Not Distributed	2869	2653
h. Total	7314	6531
i. Percent Paid Circulation	92	95
16. Total Circulation includes electronic copies.		
a. Paid Electronic Copies	4497	4567
b. Total Paid Print Copies (Line 15c) + Paid Electronic Copies	8567	8245
c. Total Paid Print Distribution (Line 15f) + Paid Electronic Copies	8942	8445
d. Percent Paid (Both Print & Electronic Copies)	96	98

✓ I certify that 50% of all my distributed copies (Electronic & Print) are paid above a nominal price.

CALL FOR PAPERS

We are always looking for good articles on Linux and the tools of the Linux environment. Although we will consider any topic, the following themes are of special interest:

- System administration
- Useful tips and tools
- Security, both news and techniques
- Product reviews, especially from real-world experience
- Community news and projects

If you have an idea, send a proposal with an outline, an estimate of the length, a description of your background, and contact information to edit@linux-magazine.com.

The technical level of the article should be consistent with what you normally read in *Linux Magazine*. Remember that *Linux Magazine* is read in many countries, and your article may be translated into one of our sister publications. Therefore, it is best to avoid using slang and idioms that might not be understood by all readers.

Be careful when referring to dates or events in the future. Many weeks could pass between your manuscript submission and the final copy reaching the reader's hands. When submitting proposals or manuscripts, please use a subject line in your email message that helps us identify your message as an article proposal. Screenshots and other supporting materials are always welcome.

Issue 194 / January 2017

Living with Windows

Approximate
UK / Europe Nov 26
USA / Canada Dec 23
Australia Jan 23
On Sale Date

Microsoft professes love for Linux, but coexistence is sometimes still a challenge. Next month we look at Windows software on Linux, open source alternatives to Windows applications, and tools for converting file formats.

Preview Newsletter

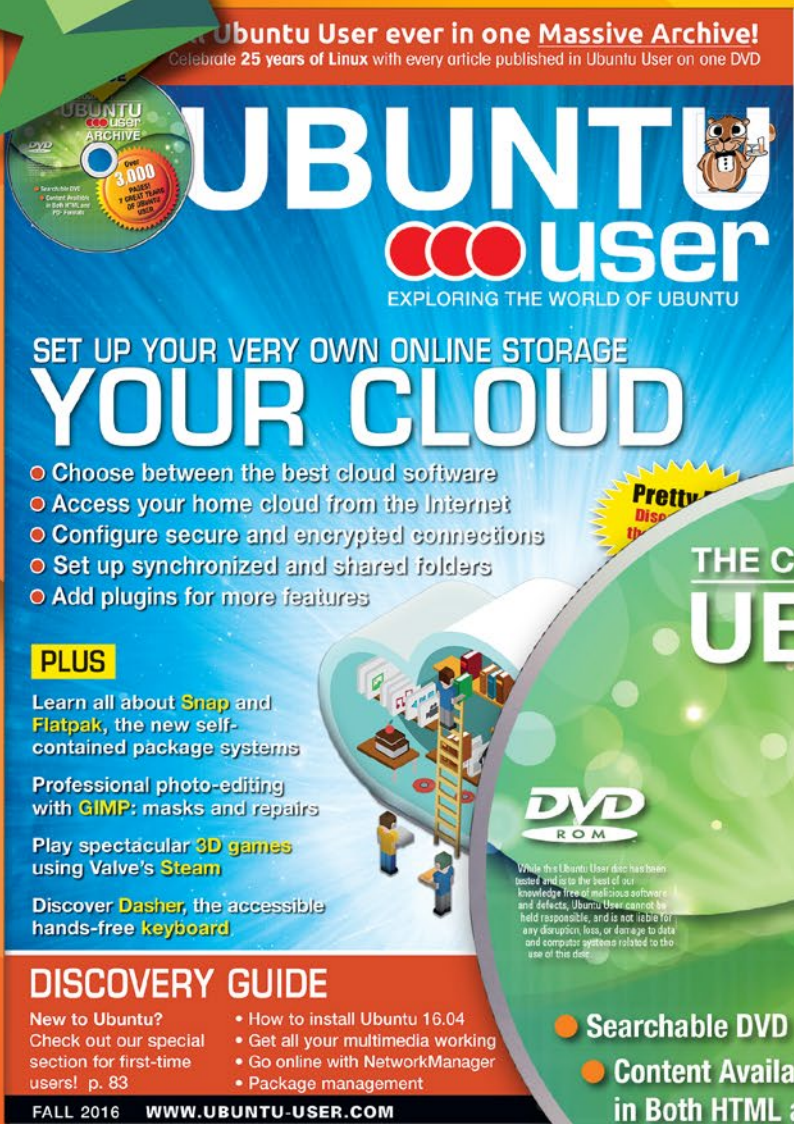
The Linux Magazine Preview is a monthly email newsletter that gives you a sneak peek at the next issue, including links to articles posted online.

Sign up at: www.linux-magazine.com/newsletter

Lead Image © Sergei Popov, 123RF.com and Jala, Fotolia.com

Celebrating 25 Years of Linux!

Get 7 years of *Ubuntu User* ever in one Massive Archive!
Celebrate 25 years of Linux with every article published in *Ubuntu User* on one DVD



UBUNTU
user
EXPLORING THE WORLD OF UBUNTU

SET UP YOUR VERY OWN ONLINE STORAGE
YOUR CLOUD

- Choose between the best cloud software
- Access your home cloud from the Internet
- Configure secure and encrypted connections
- Set up synchronized and shared folders
- Add plugins for more features

PLUS

Learn all about **Snap** and **Flatpak**, the new self-contained package systems

Professional photo-editing with **GIMP**: masks and repairs

Play spectacular **3D** games using Valve's **Steam**

Discover **Dasher**, the accessible hands-free **keyboard**

DISCOVERY GUIDE

New to Ubuntu? • How to install Ubuntu 16.04
Check out our special section for first-time users! p. 83 • Get all your multimedia working
• Go online with NetworkManager
• Package management

FALL 2016 WWW.UBUNTU-USER.COM

ORDER NOW!
Get 7 years of
Ubuntu User
FREE
with issue #30



THE COMPLETE
UBUNTU
user
ARCHIVE

DVD
ROM

While the *Ubuntu User* disc has been tested and is to the best of our knowledge free of malicious software and defects, *Ubuntu User* cannot be held responsible, and is not liable for any disruption, loss, or damage to data and computer systems related to the use of this disc.

- Searchable DVD
- Content Available in Both HTML and PDF Formats

Over
3,000
PAGES!
7 GREAT YEARS
OF UBUNTU
USER

Ubuntu User is the only magazine
for the Ubuntu Linux Community!

BEST VALUE: Become a subscriber and save 35% off the cover price!
The archive DVD will be included with the Fall Issue, so you must act now!

Order Now! Shop.linuxnewmedia.com



MicroBlade

High Density • High Performance • High Efficiency • Cost-Effective
Enterprise, Data Center, Web Applications, HPC and Cloud Computing Solutions

Intel® Xeon® Processor E5-2600 v4/v3 Product Families Supported



New!

3U MicroBlade

Up to **28 UP / 14 DP** Nodes



6U MicroBlade

Up to **112 UP / 28 DP** Nodes



MBI-6128R-T2



MBI-6128R-T2X

10Gbps

Dual Intel® Xeon® Processor E5-2600 v4/v3

New!



MBI-6219G-T7LX

HD Graphics
NVMe
10Gbps

New!



MBI-6119G-T7LX

Intel® Xeon® Processor E3-1578L v5

New!



MBI-6218G-T81X
MBI-6218G-T41X

10Gbps

New!



MBI-6118G-T81X
MBI-6118G-T41X

Intel® Xeon® Processor D

New!

SAS3

New!

New!

New!



MBI-6119G-C2



MBI-6119G-C4



MBI-6119G-T4



MBI-6219G-T

Intel® Xeon® Processor E3-1200 v5



MBI-6118D-T2/T2H



MBI-6118D-T4/T4H

Intel® Xeon® Processor E3-1200 v4/v3



MBI-6418A-T7H



MBI-6418A-T5H

Intel® Atom™ Processor



Intel Inside®.
Powerful Productivity
Outside.



Learn more at www.supermicro.com



© Super Micro Computer, Inc. Specifications subject to change without notice.
Intel, the Intel logo, the Intel Inside logo, Intel Atom, Xeon, and Intel Xeon Phi are trademarks of Intel Corporation in the U.S. and/or other countries.
All other brands and names are the property of their respective owners.