FREE DVD

Double-Sided DVD
INSIDE!

TURTLE GRAPHICS
Teach kids to program

TILING WITH REGOLITH AND I3

LINUX PRO MAGAZINE

FEBRUARY 2020

TILING

Speed up your desktop experience with Regolith and the i3 tiling window manager

GNU Stow
Simplified file management

Cryptomator
Encryption tool for the cloud

Creating desktop applications with Gnome Builder

motionEyeOS
Build a security camera with a Raspberry Pi

Unison
Sync your data over the Internet

guiscrcpy
Control an Android smartphone from a Linux system

LINUXVOICE

- Kube: Personal communication manager
- Organize your photos with Geeqie
- maddog: Using hardware emulators to explore classic assembler languages

FOSSPicks
- glChAoS.P strange attractors scout
- RawTherapee
- Qt process monitor

Tutorial
- Posting text and images with Mastodon

Issue 231
Feb 2020
US$ 15.99
CAN$ 18.50

WWW.LINUXPROMAGAZINE.COM

SHORT RIDE

Dear Reader,

This month, we who follow such things marked the passing of Unicorn Rides, a scooter vendor that billed itself as "the magical electric scooter to get where you're going." Their slogan needed some work ("get where you're going" being both tautological and overused), but the concept had some promise. Unicorn didn't actually make the scooters but, instead, sold a modified version of the popular ES2 electric scooter made by Segway. The Unicorn scooters were intended for single-owner scenarios (as opposed to on-street rental), and they were modified to include special software features. The scooter was designed to recognize the user's cell phone and unlock itself automatically when the user was near. The Unicorn app also let you authorize other users to operate the scooter, and it provided integration with the Tile device tracker platform. (Tile founder Nick Evans was one of the founders of Unicorn.)

Unicorn apparently lined up $150,000 in funding, set up a website, and started selling scooters [1]. Unfortunately, behind the promotional videos and the carefully timed social media posts, these sales were actually pre-orders for scooters that hadn't been built yet. And it turned out that, despite the extra features and marketing pizzazz, scooter buyers might have had trouble justifying the $699 price for a Unicorn when a garden-variety ES2 only costs $589. The company managed to sell a total of 350 scooters, and, suddenly, one day they ran out of money. They never met the minimum order with Segway, so the scooters were never manufactured. According to a contrite letter sent to customers "the cost of Facebook and Google ads, payments for loans, and other expenses ate through our funding faster than new orders came in." [2]

In the end, the customers received neither a refund nor a scooter. The whole enterprise played out as a massive transfer of funds, with money flowing from investors and customers to Google and Facebook and no product reaching the light of day.

In the exhilarating days surrounding the June launch, Unicorn founder Nick Evans told The Verge, "Building a quality product and lasting brand takes a lot more work than placing an order on Alibaba and slapping your new hip logo and forgettable four letter company name on the side. It's about focusing on the customers' wants and needs and finding the best way to address them. How you get there should be incidental to that goal." [3]

A noble vision, but ultimately, such sentiments are never enough if you can't pass that first hurdle of making sure you deliver a product to the customer who paid for it.

It is easy to second guess now, but startups go out of business every day. In fact, more startups go out of business than stay in business. What is strange about this one is how far they could get just based on a dream without actually delivering *anything*.

The other thing to notice is that maybe all those social network ads aren't so powerful and effective after all. It is easy to believe there is this linear relationship between the money you pour into Facebook ads and the number of customers you get. Then, of course, there is also the assumption that whoever the ad platform brings to your doorstep will buy what you are selling – that whenever your marketing department says your product is revolutionary and irresistible that means it really is and your customers will recognize that it is.

I've watched a lot of startups crash and burn from my vantage point as a salty old dog in the technical publishing field. After all I've seen over the years, I can offer a couple useful rules to those who are ready to start their own high-tech business:

- Don't believe in your own hype.
- Don't believe in anyone else's hype.

Hold fast to these simple truths, then go and follow your star.

Joe

Joe Casad,
Editor in Chief

Info

[1] E-Scooter Company Unicorn Goes Bust After Spending Big on Facebook Ads: *https://www.bbc.com/news/technology-50713723*

[2] Important Update from Unicorn's Founder: *https://www.reddit.com/r/ElectricScooters/comments/e77o38/unicorn_shutdown/*

[3] Unicorn Is a $699 Electric Scooter from the Co-Creator of Tile: *https://www.theverge.com/2019/6/20/18691357/unicorn-electric-scooter-price-subscription-tile*

WHAT'S INSIDE

Are tiling window managers still relevant for today's desktop? This month we explore the possibilities of the tiling paradigm with a modern Linux built for tiling. Other highlights include:

- **Cryptomator** – a free tool for encrypting cloud data (page 32).
- **Unison** – access your files from anywhere with SSH and this popular sync tool (page 50).

Check out MakerSpace for a story on building your own security camera with a Raspberry Pi and motionEyeOS, and turn to LinuxVoice for a look at the innovative Kube communication and collaboration client.

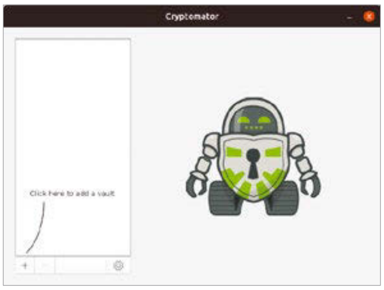
SERVICE

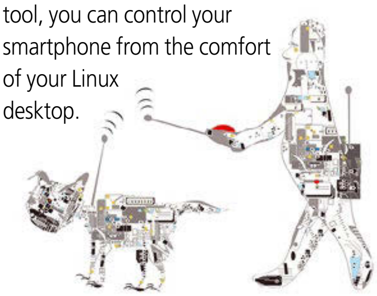
NEWS

COVER STORY

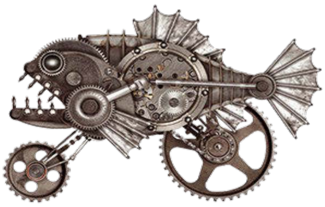
IN-DEPTH

IN-DEPTH



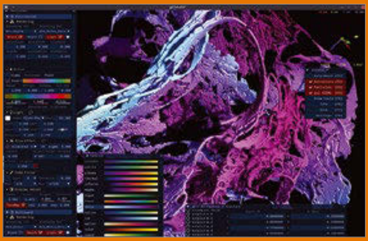


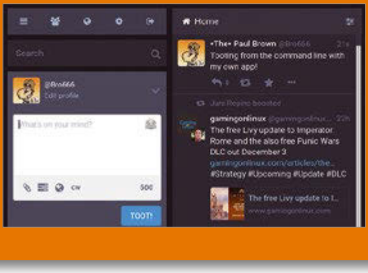
MAKERSPACE



SEE PAGE 6 FOR DETAILS

LINUXVOICE



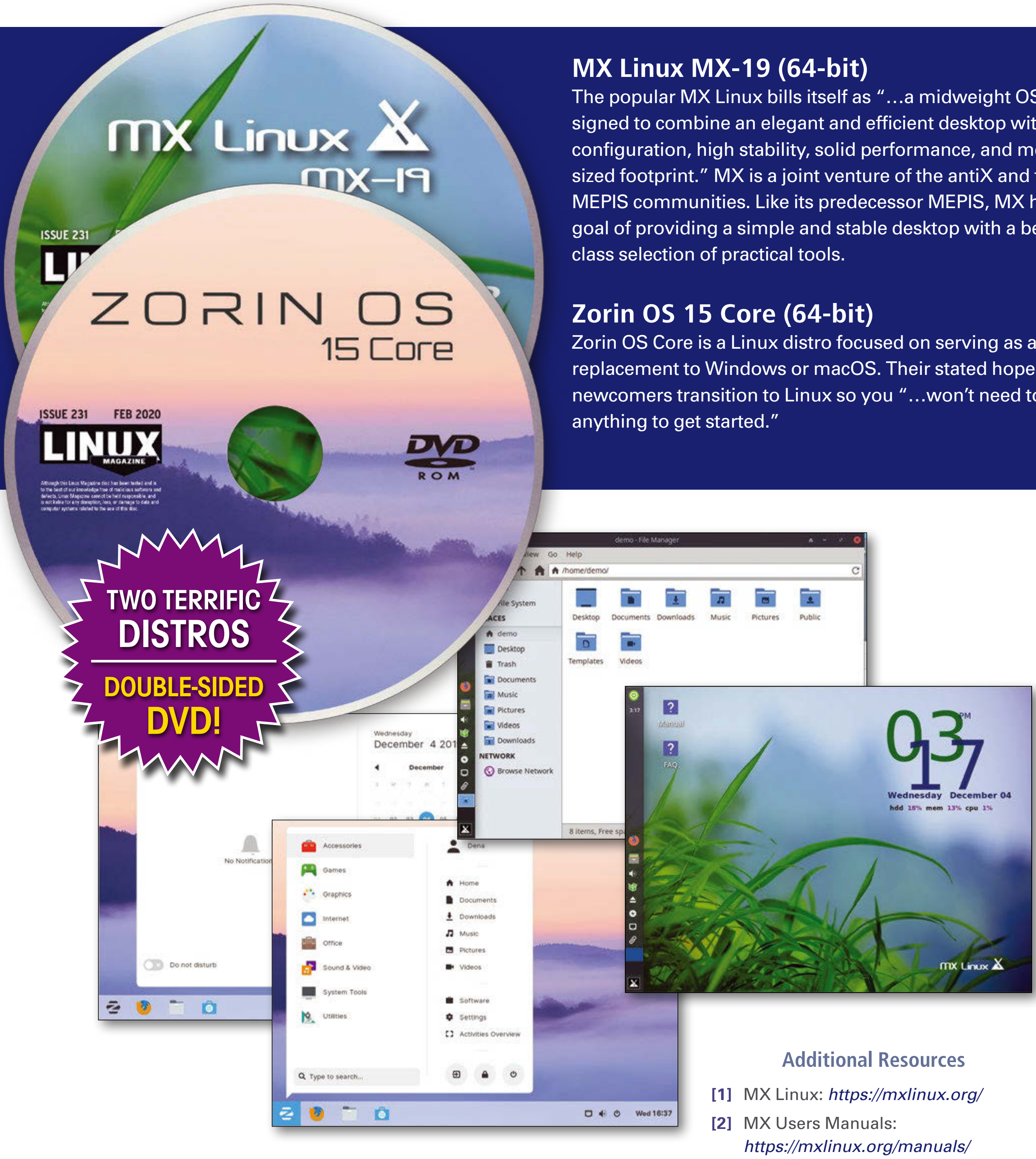


•The• Paul Brown @Bro666
Tooting from the command line with my own app!

gamingonlinux
The free Livy update to Imperator: Rome and the also free Punic Wars DLC out December 3
gamingonlinux.com/articles/the...
#Strategy #Upcoming #Update #DLC

The free Livy update to I...

On the DVD



MX Linux MX-19 (64-bit)

The popular MX Linux bills itself as "…a midweight OS designed to combine an elegant and efficient desktop with simple configuration, high stability, solid performance, and medium-sized footprint." MX is a joint venture of the antiX and former MEPIS communities. Like its predecessor MEPIS, MX has the goal of providing a simple and stable desktop with a best-in-class selection of practical tools.

Zorin OS 15 Core (64-bit)

Zorin OS Core is a Linux distro focused on serving as a drop-in replacement to Windows or macOS. Their stated hope is to let newcomers transition to Linux so you "…won't need to learn anything to get started."

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

Although this Linux Magazine disc has been tested and is to the best of our knowledge free of malicious software and defects, Linux Magazine 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.

Additional Resources

[1] MX Linux: *https://mxlinux.org/*

[2] MX Users Manuals: *https://mxlinux.org/manuals/*

[3] MX Linux Forum: *https://forum.mxlinux.org/*

[4] Zorin OS: *https://zorinos.com/*

[5] Getting Started with Zorin OS: *https://zorinos.com/help/#getting-started*

[6] Zorin OS Forum: *https://zoringroup.com/forum/*

Hone your skills with special editions!

Get to know Shell, LibreOffice, Linux, and more from our Special Edition library.

The *Linux Magazine* team has created a series of single volumes that give you a deep-dive into the topics you want.

Available in print or digital format

Check out the full library!
shop.linuxnewmedia.com

FREE DVD! LibreOffice Full Version
Includes full versions for Windows, macOS, and Linux
2019 Edition
DISCOVER LibreOffice
Free Office Suite
Find out why 75 million users have stopped paying for office software!
Edit and Save MS Office Files!
Create your own:
Word processing docs
Spreadsheets
Presentations
Databases
Use open file formats that other programs can read and import!
Also inside:
Extensions
Macros
Templates
and more!
WWW.LINUX-MAGAZINE.COM
LINUX Special 101 COOL LINUX HACKS
101 COOL LINUX HACKS
Inspirational tricks and shortcuts for Linux geeks
Repair your bootloader
Cure the Caps Lock disease
Tricks with terminal output
Disable your webcam and mic
Run C one-liners in the shell
Undelete lost files
Ignore case in file names
PHONING IN
Sync your phone with a Linux desktop
QUICK SWITCH
Change to a second distro using chroot
LINUX Special
WWW.LINUX-MAGAZINE.COM
300+ BEST BASH COMMANDS
SAVE 15% on Linux Certification See details inside
LINUX SHELL HANDBOOK
2019 Edition
SUPERCHARGE YOUR LINUX SKILLS
INCLUDES SYSTEMD BASH COMMANDS!
Travel Light
with fast and graceful keyboard commands
Power at Your Fingertips
Manipulate text strings
Pipe and redirect output
Monitor processes
Manage users and groups
Create easy automation scripts
Keep this comprehensive guide as a permanent command reference!
WWW.LINUX-MAGAZINE.COM
FREE DVD! ALL THE SOFTWARE YOU NEED!
openSUSE
THOUSANDS OF FREE TOOLS IN EASY REACH!
SWITCH TO LINUX NOW!
2019 Edition
GETTING STARTED WITH LINUX
Learn how to set up a Linux system to:
Listen to Music
Surf the Web
Play Games
and Much More!
Process Photos
MORE Powerful,
MORE Secure,
MORE Fun!
JOIN THE LINUX REVOLUTION!
LINUX Special
WWW.LINUX-MAGAZINE.COM

NEWS

Updates on tech s

THIS MONTH'S NEWS

Elementary OS 5.1 Has Arrived

If you've not heard of Elementary OS, chances are you don't know what Linux is. If, on the other hand, you have heard of Elementary OS, and you've yet to give it a try, now's a great time. Why? The latest release, 5.1 (aka "Hera") is available and it promises to be the best release yet.

Elementary 5.1 brings a number of new and exciting changes to what is often considered the most elegant desktop operating system on the market. Some of the new changes to the platform include:

- A new first-run experience that makes onboarding seamless and simple. The app is modular and includes introductions to Location services, Night Light, Housekeeping, and the AppCenter.
- A brand new login window that looks more refined and includes numerous bug fixes.
- Sideload, a new app for installing Flatpak applications that are not found in the AppCenter.
- Flatpak support in the Elementary AppCenter.
- Major upgrades centered around accessibility and the System Settings tool.

Although it might seem like a minor release (going from 5.0 to 5.1), Hera should be considered a major upgrade. Elementary releases new platform versions every two years. However, this particular release is significant enough to warrant its own name and identity.

Download Elementary OS 5.1 at *https://elementary.io/*.

Linux Kernel 5.4 Released

Linux founder Linus Torvalds has announced the release of kernel 5.4 (*http://lkml.iu.edu/hypermail/linux/kernel/1911.3/00186.html*). Included in this latest release are a number of additions and improvements that will certainly benefit desktop linux. What are the top features? Read on.

The most important addition to the Linux kernel 5.4 iteration is the new kernel lockdown mode. The goal of this new feature is a heightened level of security via the separation of UID 0 (also known as the root user) and the kernel. This feature in-

cludes both integrity and confidentiality modes. Integrity mode does not allow userland applications and services to modify the running kernel, while confidentiality mode does not allow the extraction of confidential information by userland applications and services.

As for graphics? Other important new features include support for AMD Navi 12 and 14 GPUs, as well as AMD Arcturus graphic cards and the AMD Dali and 2020 APU platforms. Intel Tiger Lake hardware now gets early support and the Nouveau open source driver sees improved display color management.

Finally, file systems are getting a couple of important additions. First off, an optional case-insensitive file and folder handling feature has been made available in F2FS. This will bring support for case-insensitive file-name lookups. With this feature, F2FS case folding can be set on a per-directory basis. Second, exFAT support is now available for Linux. With this new option, the transferring of files bigger than 4GB will be possible with drives formatted to the exFAT file system.

System76 To Design And Build Laptops In-House

System76 (*https://system76.com/*) has been selling laptops and desktops, preinstalled with Linux, since 2005. But up until recently, their desktop machines were designed by third-party OEMs. That all changed in 2018 when System76 began selling their in-house designed and built desktops, the Thelio (*https://system76.com/desktops*). The Thelio machines have been met by glowing reviews around the globe, and can be specced high enough to accommodate serious loads.

As for laptops? System76 has been depending on Sager and Clevo to design their hardware. These laptops have served the company (and consumers) well. But once System76 proved they could build one of the finest Linux-powered desktops on the market, they set out to discover if it was possible to also produce laptops in house.

© Robertas Pezas, 123RF

Starting in 2020, the company will begin the journey to make this dream a reality. But these won't just be the standard fare. According to Carl Rochelle, CEO of System76, the laptops will follow in the same footsteps as the Thelio and deliver as much form as it does function. In other words, the company plans on designing a laptop that can stand toe-to-toe with both the aesthetic and performance (relatively speaking) of their desktop offering.

The in-house designed and built laptop production should take 2 to 3 years to get on its feet. Until then, System76 will continue to offer the current laptop lineup, which includes the Galago Pro (*https://system76.com/laptops/galago*), Darter Pro (*https://system76.com/laptops/darter*), Gazelle (*https://system76.com/laptops/gazelle*), Oryx Pro (*https://system76.com/laptops/oryx*), Adder WS (*https://system76.com/laptops/adder*), and Serval WS (*https://system76.com/laptops/serval*).

The PinePhone Pre-Order has Arrived

Created by Pine64 (*https://www.pine64.org/*), the PinePhone is an affordable Linux phone with a price tag of only $149. This phone is targeted at Linux enthusiasts and developers looking for privacy-centric open source software and hardware kill switches.

MORE ONLINE

Linux Magazine

www.linux-magazine.com

ADMIN HPC

http://www.admin-magazine.com/HPC/

Profiling Python code
- **Jeff Layton**

Profiling Python code – as a whole or by function – shows where you should spend time speeding up your programs.

Linux device mapper writecache
- **Petros Koutoupis**

Kicking write I/O operations into overdrive with the Linux device mapper writecache.

GPU Computing

http://www.admin-magazine.com/HPC/

Exploring AMD's ambitious ROCm initiative
- **Joe Casad**

Three years ago, AMD released the innovative ROCm hardware-accelerated, parallel-computing environment.

Porting CUDA to HIP
- **Joe Casad**

You've invested money and time in writing GPU-optimized software with CUDA, and you're wondering if your efforts will have a life beyond the narrow, proprietary hardware environment supported by the CUDA language. Welcome to the world of HIP.

ADMIN Online

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

Google Cloud Storage for backups
- **Thomas Drilling**

We compare Google Cloud Storage for Internet-based backups with Amazon S3.

Credential harvesting at the network interstice
- **James Stanger**

To thwart credential harvesters at the network interstice, you must understand how attackers exploit browser transactions.

The new OpenShift version 4
- **Martin Loschwitz**

Red Hat launched the brand new OpenShift 4 with a number of changes that might suggest upgrading or even getting your feet wet if you've stayed out of the pool so far.

The specs for the PinePhone are humble (to say the least). The device includes an Allwinner A64 1.2 GHz quad-core A53 CPU, 2GB of RAM and 16GB of storage, a 5.9" IPS LCD display, a 2MP front-facing camera and a 5MP rear-facing camera, a Mali 400 MP2 GPU, a 3000 mAh battery, and a USB C port.

© Cienpies Design, 123RF

The one feature that will excite privacy advocates is the inclusion of hardware kill switches for Wi-Fi, cellular, camera, speaker, and mic. For any mobile user concerned about privacy and security, a kill switch allows quick disabling of a feature. And for those who like to experiment with different flavors of Linux, the PinePhone will eventually support the installation of other distributions, such as Ubuntu Touch, PostmarketOS, LuneOS, Nemo Mobile, and Aurora OS/Sailfish OS. Because there has been so much support from the Linux community, it is possible the PinePhone will eventually be available with a number of operating systems to choose from. When the PinePhone is released to the general consumer, it will most likely ship with a version of KDE Plasma, which has been optimized for the smartphone form factor.

The first release of the PinePhone is called the "Braveheart Edition." This release is strictly for enthusiasts and should not be considered a finished product for the general public. The "Braveheart Edition" goes up for pre-order on November 15, 2019. Pine64 has targeted March 2020 for the general release.

SUSE Adds SUSE Linux Enterprise to the Oracle Cloud Infrastructure

SUSE (a Gold-level member of Oracle PartnerNetwork) recently announced that SUSE Enterprise Linux is now a part of Oracle Cloud Infrastructure (*https://www.suse.com/c/news/suse-brings-enterprise-linux-to-oracle-cloud-to-meet-growing-demand-for-cloud-based-business-deployments/*). This move also brings Oracle into the ever-growing membership of the SUSE Partner Program for Cloud Service Providers.

Both SUSE Linux Enterprise and SUSE Linux Enterprise Server for SAP Applications will allow customers to leverage high performance virtual machines and bare metal compute for Linux-based workloads. According to Naji Almahmoud, SUSE vice president of Global Alliances, "SUSE's collaboration with Oracle Cloud Infrastructure allows us to meet growing customer demand for the agility and cost benefits of cloud-based business-critical applications…"

Via a bring-your-own-subscription arrangement, SUSE customers will be able to transfer existing SUSE subscriptions to Oracle Cloud Infrastructure so they can deploy new workloads or migrate existing workloads from their current datacenter to Oracle Cloud. There will be no additional cost imposed by SUSE for this new addition, and customers will be able to continue taking advantage of their existing relationship with SUSE support.

Vinay Kumar, VP of Product Management, Oracle Cloud Infrastructure, said, "SUSE Linux Enterprise Server on Oracle Cloud Infrastructure offers enterprises more choice as they transition to the cloud." Kumar added, "Oracle and SUSE have a common goal of providing open and reliable infrastructure to support our customers' digital transformation."

11-12 March 2020
ExCeL, London

CLOUD EXPO EUROPE

MISSION IMPOSSIBLE

Cloud Expo Europe, the UK's biggest and best attended technology event, returns on 11-12 March 2020 at ExCeL London.

Technology enabled change is on the boardroom agenda for businesses of all types and sizes. Cloud Expo Europe is the UK's leading event for connecting technologists, business leaders and senior business managers with experts, solutions and services to help accelerate digital transformation plans.

Whether you are cloud-first, scaling up, refining, or just getting started, Cloud Expo Europe is an unrivalled opportunity to meet with leading technology innovators and service providers. Network with your peers. Access a wealth of knowledge and advice including emerging trends, tech deep dives, lessons learned and market forecasts.

Our 2020 speaker line-up includes:

DENISE DOURADO
Cloud Digital Transformation Director
HMRC

DANIEL MARION
Chief of Information and Communication Technology
UEFA

ANNA BARSBY
Award-winning CIO

CIJO JOSEPH
Chief Information Officer
Mitie

MAJOR-GENERAL THOMAS COPINGER-SYMES
Director of Military Digitisation
Joint Forces Command

NICK MITROVIC
Chief Technology Officer
Oxfam

Visit www.cloudexpoeurope.com for more information or contact the team today to discuss further on +44 (0) 207 013 4999

Co-located with: DEVOPS LIVE | CLOUD & CYBER SECURITY EXPO | SMART IOT | BIG DATA & AI WORLD | BLOCKCHAIN TECH WORLD | DATA CENTRE WORLD

ORGANISED BY CloserStill

Zack's Kernel News

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

By Zack Brown

When to Break the ABI

It's rare that a patch sneaks through the development process and changes the Linux kernel application binary interface (ABI), though it's common enough to see a patch that tries to do it or to see a developer that advocates doing it. When it has occurred, Linus Torvalds has always made it very clear that virtually nothing short of a security hole could possibly justify such a thing. But we almost never see a case where a patch is actually accepted into the kernel, and then it is discovered that it changed the ABI.

To understand the ABI in context, consider the kernel application programmer interface (API). The API is a set of library routines that can be referenced by user source code in order to give commands to the kernel or get information out of the kernel.

Linus has no trouble at all changing the kernel API. Well, he has certain standards, but there is certainly no interdiction against it; it's just a normal part of kernel development. The reason is that in order to run into problems using the kernel API, a user must be trying to compile the source code of another program that uses that API. In that case, if the user sees the problem, they can patch the source code themselves by hand, using whatever programming language their code is written in. Once their source code is updated, they can compile and run it, no problem.

The ABI is very similar to the API – but instead of representing a source code library, it represents the actual compiled machine code interface, the CPU opcodes themselves that form the library call. For a user to run into a problem with the ABI, they must be trying to run software that has already been compiled to use that particular ABI. In such a case, the user may not have any source code to patch; they may simply have a closed-source blob of executable code that they've been using for years and can never fix or replace.

That's the problem. If it were possible to freely fix or replace binary executables when they ran into ABI conflicts, Linus would certainly not care nearly as much about breaking the Linux ABI. In that case, he would probably be just as willing to break the ABI as he is to break the API – the rationale being that it's OK so long as no actual users are inconvenienced.

But Linus is fanatically loyal to existing binary code. He refuses to allow the progress of Linux kernel development to break binary code that is already running in the wild. Anyone, anywhere, who is relying on compiled software, must be able to upgrade their kernel and continue to run their existing binary user code, without fear. As I say, virtually the only exception to this rule is where security is concerned. Linus would hew off any piece of the Linux kernel no matter how useful, if that were necessary to eliminate a security hole. Above all else – even functionality – Linux is a secure system.

The ABI policy was demonstrated recently when Douglas Anderson from the Chromium project posted a patch to revert a cryptography feature, on no other grounds but that the earlier patch had broken the kernel's ABI. He said, "The commit made a clear and documented ABI change that is not backward compatible. There exists userspace code that relied on the old behavior and is now broken." He gave a link to the part of the Chromium project that relied on the old ABI.

It's important to note that even though Chromium is an open source project, it still represents compiled executables that rely on the binary interface. Merely having source code available for a given project is not enough to justify breaking the kernel's binary interface relied on by that project.

Douglas was even willing to deal with an ABI change – you never know

Author

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**.

when it might actually turn out to be a legitimate security issue – but he said the Chromium developers would strongly desire some help managing the necessary changes to their codebase in that case.

Eric Biggers thanked Douglas for the report and asked to know how this broke Chrome OS. He also remarked that if it were really necessary to revert the patch, then the kernel would need to document the fact that a particular error code, which could arrive when doing encryption, was actually a filesystem-specific error. Reverting the patch would essentially make the kernel's behavior more difficult to understand and explain, which as Eric put it, "we'd really like to avoid…."

Eric pointed out that an alternative patch would break the ABI in a different way. He asked if Douglas was proposing that the kernel keep the complex, difficult-to-explain behavior.

Douglas acknowledged that he didn't really know enough about this particular area of the kernel. He said, "I guess I'd have to leave it up to the people who know this better. Mostly I just saw this as an ABI change breaking userspace which to me means revert. I have very little background here to make good decisions about the right way to move forward."

Eric swallowed that, though it must have been difficult, and agreed that it looked as if the patch did indeed need to be reverted. He proposed that for the next kernel version, the existing behavior should be broken equally across all filesystems, so that the behavior would at least be consistent. Yes. The feature proposal is to break the rest of the filesystems in the same way, rather than accept even a single ABI change. As Eric put it, "I think we should try to make things slightly more sane by removing the same check from f2fs and fixing the documentation, so that at least each ioctl will behave consistently across filesystems and be correctly documented."

In addition to this, Eric had to swallow the equally bitter pill of trying to track down exactly where the ABI breakage truly was. After all, the original patch existed for a reason, and that reason still needed to be addressed if possible, but just without the ABI breakage.

But there's a twist! As Eric proceeded to unravel the true nature of the breakage, he found a way to repair the Chromium behavior without actually reverting the patch reported by Douglas. In response to this, Guenter Roeck of Google suggested that the kernel maintainers hold off on accepting the reversion. He said, "I'll do more testing next week, but as it looks like it may no longer be needed, at least not from a Chrome OS perspective."

And Theodore Y. Ts'o, very high up in the kernel developer hierarchy, agreed! He said he would wait on sending the reversion to Linus.

That was the end of the discussion, but it's very interesting. Apparently the status of an ABI breakage can be influenced by whether there are any actual users of that particular interface anywhere in the world. Having fixed the problem from the Chrome OS perspective, the kernel developers had found a sort of reprieve. If no other users come forward to complain about this particular breakage, it may even be conceivable that Linus would allow the ABI breakage to persist.

So the policy towards ABI breakage appears to be that it's OK to solve a security problem or if there are no actual users of the ABI in question.

Status of vboxsf

There was an interesting interaction between the Linux maintainer (Linus Torvalds) and the maintainer of the stable branch (Greg Kroah-Hartman). Linus recently accepted a patch to a kernel release candidate (-rc), which brought the VirtualBox Shared Folder (`vboxsf`) filesystem into the staging area of the kernel source code. Specifically it was Linux 5.4-rc7, relatively late in the process towards the official 5.4 release. The staging area is traditionally a place where upcoming features can get the widest possible distribution. They aren't actually in the compiled kernel, because the staging area is kept isolated. But they are in the source tree, and anyone downloading that tree also gets the staging area. This is very useful for getting as many eyes as possible on new features, before they are actually transitioned into the main kernel code and put into actual use.

It was Greg who fed the patch up to Linus at that time, and he got the patch from Christoph Hellwig, who had said it was ready to go into the kernel. Greg apparently misconstrued what Christoph had meant and put the new filesystem into the staging area instead of the main kernel for actual use. So after Christoph pointed out Greg's mistake, Greg submitted a new patch to Linus, for the -rc8 kernel, migrating `vboxsf` out of the staging area and into the kernel proper.

Linus replied definitively:

"No.

"I was unhappy about a staging driver being added in rc7, but I went 'whatever, it's Greg's garbage'.

"There is no way in hell I will take a new filesystem in rc8.

"Would you take that into stable? No, you wouldn't. Then why is this being upstreamed now.

"Honestly, I think I'll just delete the whole thing, since it shouldn't have gone in in the first place. This is not how we add new filesystems."

Greg accepted this gracefully, saying, "Fair enough, sorry for the noise."

However, Hans de Goede complained, saying:

"The problem is that Al Viro, after an initial review around v2 or v3 of the patch, which I believe I have fully addressed, has been ignoring this patch/new fs for over a year now. I've pinged him repeatedly both via email and on irc, but with no luck. I guess he simply is too busy with other stuff.

"I did ask other fs developers to review and have gotten reviews from David Howell and Christoph Hellwig. I've addressed all their review remarks and I've had reviews of the newer versions with just a few nitpicks remaining. I've also addressed those nitpicks. But I never got an Acked-by or Reviewed-by from either of them on any of the newer versions.

"I even talked to various people about this at plumbers, but I did not get any traction there either.

"On the advice of Christoph I've asked Andrew Morton to take this directly under fs/ instead, twice. When this all went no where I went the staging route, with the current result."

Linus did not reply directly, but when he announced -rc8 he did say, "The other noticeable thing in the diffs is the removal of the vboxsf filesystem. It will get resubmitted properly later; there was nothing obviously wrong with it technically; it just ended up in the wrong location and submitted at the wrong time. We'll get it done properly probably during 5.5." Regarding the lateness of submitting `vboxsf` for inclusion, he remarked, "I considered just making a final 5.4 and be done with it, but decided that there's no real downside to just doing the rc8 after having a release cycle that took a while to calm down." So clearly the -rc7/-rc8 time frame was simply too late this time around.

For me, there are several interesting points to this whole exchange. The use of the staging area is always interesting, because it was created specifically to solve the problem of new features failing to get enough testing before going into the main tree. The idea of having a tree within the tree, to make sure upcoming features were as available as possible, was a simple and surprising innovation.

And in this case, although Linus acknowledged that the filesystem looked technically OK and had already been accepted into the staging area, he pulled the entire patch out again in the very next release candidate, just because the submission itself hadn't been done properly. This seems to indicate a general principle of keeping things orderly, in the face of the tremendous onslaught of new features that pour into the kernel with every release cycle. Linus apparently wants to make it completely unambiguous to developers exactly how each new feature should proceed on its way into the tree.

The other thing I find very interesting about this is Hans's attempt to do an end-run around developers who didn't seem to be responding fast enough – particularly Alexander Viro, who comes as close as it gets to sitting side by side with Linus in the developer hierarchy. Hans tried various ways to get Al's attention, raised the issue at developer conferences, and then followed other well-placed developers' advice to bring Linus's attention to the project. And lo and behold, it's a near certainty that `vboxsf` will get solid and timely consideration before the release of Linux 5.5. ■■■

The Linux Foundation

Promoting Adoption

As the Linux Foundation approaches its 20th anniversary, we sit down with Jim Zemlin to talk about how the nonprofit has expanded its mandate since its inception. *By Mayank Sharma*

In 2000, the Linux Foundation arose from a merger of Open Source Development Labs (OSDL) and the Free Standards Group (FSG) to work towards standardizing Linux and promoting its adoption. Recently, we caught up with Jim Zemlin, the Linux Foundation's long serving executive director, following his keynote at the Open Source Summit Europe 2019 [1] in Lyon, France, to discuss how the nonprofit has matured through the years.

Linux Magazine: What does the Linux Foundation do? Not through its various groups, but what does the foundation itself do?

Jim Zemlin: So there's a few ways to think about the foundation where we are the infrastructure that allows these projects to be a good upstream for a lot of, in particular, commercial downstream usage. And that involves managing the intellectual property. So we have several attorneys on staff who manage the governance, the intellectual property of those projects.

This year alone, we will have people sign 33,000 contribution agreements. And so just the processing and management of all the intellectual property agreements and frameworks around these projects is a lot of work.

We are an event organization. This [Open Source Summit Europe] is one of 150 events that we will run this year. The biggest event will be about 12-13,000 people in November.

LM: Open Source Summit Europe isn't the biggest?

JZ: No this isn't even close. Our KubeCon event in November [2] will be [the biggest]; I think it'll be somewhere probably between 12,000 and 13,000 people. And then we have small hackathons. And then this [Open Source Summit Europe] is sort of a midsize.

We are an e-learning organization. So we have an entire training team that creates OpenJS certification that provides free learning tools and curriculum for people from all over the world to learn about Linux and Kubernetes and all of our projects.

We promote all of these projects; we're the manager of the IT infrastructure for a lot of these projects. So making sure that in addition to projects that use GitHub or GitLab, we have, you know, CI/CD frameworks and all the tools that you would need to release software and do that in a secure way.

We do research; we're a research organization. One example is we're partnering with Harvard University to try and understand what the world's most important software is from a security and usage perspective and understand how people behave in those projects and how we could help improve their security. Those are the fundamental things that we do for all the projects that we promote.

LM: How has the Linux Foundation grown over the years? Is there something that you're doing now that you would not have thought of doing a decade ago?

JZ: I'll tell you a story [about] my first date with my now wife. She is a tech executive herself. She's a partner at PricewaterhouseCoopers and went to Harvard Business School, a very Type A person. So this is a blind date. She asked me what I do for a living. And I say I work at this nonprofit, and it's open source. It was like 16 years ago. The look of disappointment was just palpable. She was checking her watch, like "I must get going."

Fortunately I convinced her to be my wife. But I think back then a lot of open source was about evangelizing the concept of open source – that it was safe and a better model for innovation. And I don't think we ever thought that Linux would be as successful as it is. I don't think we were ever constrained in our thinking. But I mean, it's just now the most ubiquitous and successful software in the world. I wouldn't have thought that the open development model would leap into adjacent areas around standards setting and data, the things I talked about this morning [in the keynote]. So, you know, those were things where we, to some degree, stumbled upon them. But in many ways – and I think Linux is the best example – there's no roadmap for the Linux kernel, right? It sort of works with how the market and the world is evolving. We've been similar in that way, but we have a set of core principles around having projects that are open for anyone to participate in; [you] don't need to join any organization to do that. We believe in open technology. And those are the things that guide us as we choose new opportunities to work on.

LM: I was just reading about the Academy Software Foundation [3], and you mentioned in a talk that it took you two years to get it up and running. Is that typical for a project? What are the conversations that happen behind closed doors?

JZ: At the Academy Software Foundation, Rob [Bredow], who I mentioned in my talk, is the president of ILM [Industrial Light & Magic] – I think he was the CTO of Lucas at the time. [He] convened a group of the leading technologists from all the major film studios at the Academy of Motion Picture Arts and Sciences down in Los Angeles. It was a cool meeting, because it was in the boardroom of the Academy. These are the people who created the Oscars, so there's a mystique to it. And this was pretty early in that they understood that by sharing the maintenance and development of this underlying infrastructure technology, they could improve things for everyone. But that it is also an industry [where] – because similar to software – films are based on intellectual property, and trademarks and copyright that are highly protected and are, you know, sort of how they make money.

So it took a very long time, not to convince the technologist, Rob – I mean he's just a engineer's engineer – but the lawyers took us a couple of years to convince that this was a safe framework and that all those competing studios could work together in a safe way that would safeguard both their intellectual property and allow for the open collaboration.

So that is probably 80 percent of what I do is work with a lot of folks who may not understand open source software or how to do it, but have understood through looking at other successes that this would be a good way for them to work with their industry. And, you know, we've done that in telecommunications here in Europe in the energy sector with our LF Energy project [4]. We've done it in the financial services world with our blockchain initiative, Hyperledger [5]. And we spend a lot of time working to convince the decision makers and stakeholders in these big entities that it's okay to open their source code – that these sharing regimes are safe and reliable. And that just takes a lot of time.

LM: And how do you decide on an area or aspect of computing that could use the Linux Foundation treatment?

JZ: We have a huge amount of inbound interest of entities who come to us and ask us to help them understand how they can use the methods and the tools that we have to help them in a particular area of technology, whether it's cybersecurity or, you know, in the case of the utility sector with energy. Recently we've been having conversations with some life science companies. Governments often approach us and ask us how they could use open source software to create better outcomes for their citizens, and we do spend a lot of time filtering through those requests to make sure that anything we work on will have a big impact that adheres to our open principles and [whether] we have the resources to support them. Those are kind of the ways we think about choosing what to work on.

LM: Do you make requests to other parties to get together and work on something?

JZ: I haven't in awhile. We were not necessarily out there soliciting. I think there was a period of time where we looked at, for example, a vertical view of the software stack and asked ourselves as an organization, "Ok, what components of the stack are still kind of closed and proprietary places where there's an opportunity for open source?" And what's crazy to say is that if you look at [it] just sort of from the application layer with things like Node.js all the way down to the OpenBIOS project, open hardware projects, and from the operating system, networking stack, and everything in between, today, [for] much of the modern software industry, that entire stack is being driven by open source software. So that vertical view of the world has been something where we've, I think, seen some success. But we, a very long time ago, thought of the world that way and then intentionally worked with folks who might have an opportunity at various layers.

Today, we much more think about sort of how we would work with wholesale industries. So again, the energy sector is a great example of where most modern utility operators are in state run organizations. The grid operators themselves are private companies. But they don't have necessarily a ton of incentive to create proprietary software around certain aspects of that industry. And so this is an area that's just ripe for being able to use open source to get better outcomes for their customers, which are you and me and anyone who uses the electric grid.

And so those are areas where we will work with big industry stakeholders and, in the case of the energy industry, that'll impact climate change and can help improve the world. That's the final thing [that] we still continue to think about, which is if we're not having some kind of impact on important issues, then why bother? Go work for a for-profit company.

LM: Although you are a nonprofit, do you need to make money to fund events?

JZ: So the other thing I do besides what we just discussed is I'm a fundraiser. So I go around and work with different companies who depend on open source to underwrite the organization. Fortunately we provide a lot of value, whether it's through these events where people sponsor it or through our open source projects that these companies depend on. So explaining the value and getting people to help support them has not been incredibly challenging, but it's certainly not easy either.

LM: So, it's all funding and collaborations. Are there any commercial agreements or fee-for-service type of arrangements?

JZ: We get asked all the time, "can you consult with us to teach us how to leverage open source?" We're not a consulting company. We don't want to do that kind of work.

LM: You're not the only open source foundation out there. For lack of a better word, what's your unique selling proposition (USP)? I got a press release recently that said that Databricks just migrated their Delta Lake platform to the Linux Foundation. What would compel a project to come under the guidance of the Linux Foundation?

JZ: Oh, we're the largest organization of our kind. We can work with projects from large to small; we have an entire infrastructure, event, training, [and] legal governance team. We really have, by far, the most comprehensive support infrastructure around us. But having said that, there are certainly many good open source organizations, many of which we also support, whether it's financially or through working with them to cohost events or other things.

LM: Not everything in this world is powered by open source software. What's the one area or aspect of computing that bugs you, because it isn't open source?

JZ: I'd love to see more open source in the life sciences and pharmaceutical industry. I think that's an area where it would – as you have more computational challenges around introducing new drugs and so forth – help accelerate that market and make it more efficient. And that's a market we all care about. It's an industry that is obviously very conservative when it comes to intellectual property. And I'd just love to see more there, because healthcare is something I think we can all agree upon [that] is important.

LM: Any particular initiatives?

JZ: None that I could talk about right now.

LM: Fair enough. You introduced the KernelCI initiative [6] in your keynote today. What's that about?

JZ: So it's a common framework for testing in the kernel that will help to improve the way that upstream testing happens in the Linux kernel project. It will expand test coverage in a project that's critical to all of us. And essentially help us identify and remediate bugs in the kernel that often become security vulnerabilities, and so it's just in everybody's interest to do that. I'm super excited about it; I've been working on [it] for years, so I expect big things from that project. You can talk to Greg [Kroah-Hartman] about it, but I think he would admit that testing in the kernel isn't the best, in that improving that is incredibly important as the kernel is more important to all of us, and KernelCI is [a] long time coming. And good for those guys to have gotten there.

LM: What has the job taught you? How has it helped you grow as an individual?

JZ: Yeah, I mean I publicly talk about that. I tell this to everyone who considers even working at the foundation, which is the number one thing it teaches you is emotional patience. And the medium for making critiques in open source is a version control system where it's a pull request or critique. And so, often at a conference like this, I'll have someone come up to me and say, "Hey, Jim, I have a quick question for you." And I'll respond to say, "Well, is it really going to be a question?" They'll say, "Well, it's more of a comment." I am like "Is it a criticism?" and often that is the case.

But if you're patient and a good listener, I think that this is a job where you can succeed and that every year that's more reinforced, and I think as the world becomes more interconnected that people who succeed will be those who have those emotional skills to be collaborative, patient, understanding, [and] empathetic, and we try and teach everybody at our organization to behave that way. ■■■

Info

[1] Open Source Summit Europe 2019: *https://events19.linuxfoundation.org/events/open-source-summit-europe-2019/*

[2] KubeCon 2019: *https://events19.linuxfoundation.org/events/kubecon-cloudnativecon-north-america-2019/*

[3] Academy Software Foundation: *https://www.aswf.io/*

[4] LF Energy project: *https://www.lfenergy.org/*

[5] Hyperledger: *https://www.hyperledger.org/*

[6] KernelCI: *https://kernelci.org/*

Author

Mayank Sharma is a technology writer. You can read his scribblings in various geeky magazines on both sides of the pond.

Exploring the i3 tiling window manager with Regolith Linux

Timely Tiling

The i3 window manager organizes applications in a flexible grid. Regolith is an Ubuntu-based Linux alternative that makes it easy to explore i3 and window tiling. *By Christoph Langner and Joe Casad*

Operating a computer is a bit like driving a car. You sit behind the wheel (in front of the keyboard and mouse) and drive off. You know what to do regardless of the car's manufacturer or model. With a computer, users are accustomed to a similar operating experience regardless of the manufacturer or even the operating system. Central elements such as the desktop, windows, folders, and files are largely the same, whether you are working in Linux, macOS, or Windows. This dominant style of user interface is known as the WIMP format (Windows, Icons, Menus, Pointer), and it was originally developed by Xerox Corporation, which had a GUI-based computer operating system as early as 1973 and released the commercially viable Xerox Star in 1981 [1].

Although mouse pointing and window pushing are standard practice around the world, a defiant minority of users seeks to keep mouse interaction to a minimum. Many of these users swear by command-line tools like the Vim editor or Emacs, file managers like Ranger, or text-based mail clients like Mutt. But you don't have to do without a graphical desktop environment. Tiling window managers organize application windows without the need to continually resize them and move them around. You can navigate to applications on the screen by means of intuitive keyboard shortcuts – without reaching for your mouse. Proponents of tiling window managers believe that, with a little practice and dedication, you can work far faster with a tiled desktop than with a conventional desktop environment.

Tiling Options

Window tiling is not a specific product or a tool but is more like a philosophy. Many different operating systems support tiling in a number of ways. Several versions of Microsoft Windows have a built-in option for tiling. Both KDE and Gnome provide a number of different extensions that offer tiling support [2] [3].

Or, if you'd rather avoid the bloat of Gnome or KDE, the Linux environment supports a number of other desktops and window managers that were designed from the ground up to support tiling.

i3 [4] is a classic tiling window manager for Linux. You will find i3 in the package sources of all common Linux distributions. The current version at the time this article was written was 4.17.1, which was released at the end of August 2019. i3 offers many advanced features, such as virtual workspaces. In addition to basic tiling, i3 also lets you stack windows or bundle them into tabs.

You can install i3 on your Linux system anytime and configure your X-based Linux graphics environment to support tiling. (Another window manager called Sway bills itself as a "drop-in replacement" for i3 in Wayland graphics environments [5].) However, many users would prefer to avoid the complication and disruption of reconfiguring, adapting, and optimizing a new window manager on their existing system.

The success of Ubuntu, with its many flavors (Kubuntu, Lubuntu, Xubuntu), is an indication of the public's preference for systems that *just work* with the desired desktop environment already built in and optimized. Why not a full-featured Ubuntu system that supports tiling out the box?

Introducing Regolith

Regolith [6] jumps into the breach, offering a fully equipped Ubuntu-based system preconfigured for tiling with i3. The current version of the Regolith R1.2 system is based on Ubuntu 19.04. The Regolith developers have successfully combined i3 with the graphical system tools of Gnome, resulting in a lean desktop with keyboard-optimized controls and convenient tools for system administration.

If you're considering making the switch to a tiling environment, or if you are just curious about how tiling would look with the familiar Gnome/Ubuntu toolset, try Regolith.

Installing

To install Regolith on your computer, download the 2.1GB ISO image [7]. Since Ubuntu no longer offers a version for 32-bit computers, Regolith cannot offer a 32-bit version either. Burn the image on a DVD or put it on a USB stick with at least 4GB capacity. At boot time, you can choose whether to explore the system in live mode or install the operating system on your computer's hard drive. The Regolith developers also offer another interesting option: You can convert your existing Ubuntu system into Regolith by installing a predefined collection of additional packages (see the box entitled "Ubuntu Roots").

Regolith installation is similar to Ubuntu installation. Access partitioning tools as needed and choose whether to install a minimal or complete system. The easy installation ensures that even newcomers can set up the system without difficulty. Thanks to the solid Ubuntu underpinnings, Regolith reliably detected all the hardware in our tests. After completing the initial installation, you will need to reboot. After you log in, you are taken to a welcome dialog in the Gnome login manager, where you can set up the keyboard layout, enable positioning services, and integrate existing online accounts with Google, Nextcloud, Microsoft, or Facebook.

Regolith software is much like the typical set offered with other Linux distributions. You'll find the Firefox web browser, along with LibreOffice, the Shotwell photo manager, Simple Scan, and other favorites. You have a choice of two terminals: Simple Terminal (aka st) is the default terminal, but you can also choose the Gnome terminal (dubbed *Regolith Terminal* in Regolith).

Exploring

Regolith and tiling window managers were created for people who would rather stay on the keyboard and not go searching for a mouse. The key to working efficiently in Regolith is to memorize the keyboard shortcuts that let you open and close applications, move between windows, and resize windows for an efficient workspace. See Table 1 for a list of common shortcuts. Note that all the key combinations begin with the Super key. On most computer keyboards, the Super key is the key with the Windows logo. (This key is sometimes called the "Windows key," but as you can imagine, that name is not popular with Linux users.) On Apple keyboards, the Super key is the command key. The Super key gets its name from the X11 nomenclature, where this key sets the shift bit designated as *Super*.

Ubuntu Roots

Calling Regolith an independent distribution is an exaggeration: The system relies entirely on the Ubuntu repositories and includes only one additional PPA with the system configuration and a few additional packages [8]. In fact, it is possible to convert an ordinary Ubuntu system into a Regolith system. The developers support Ubuntu 18.04 "Bionic Beaver," as well as Ubuntu 19.04 "Disco Dingo."

To turn your Ubuntu system into Regolith, open a terminal window on "Bionic" or "Disco" and type the commands from Listing 1. The first command integrates the PPA package source provided by the Regolith developers. Then refresh the package manager and install the software packages required for Regolith. The system automatically retrieves the i3 window manager from the Ubuntu repositories. After the install, log out and, in the gear menu, select Regolith as the desktop to load in the login manager.

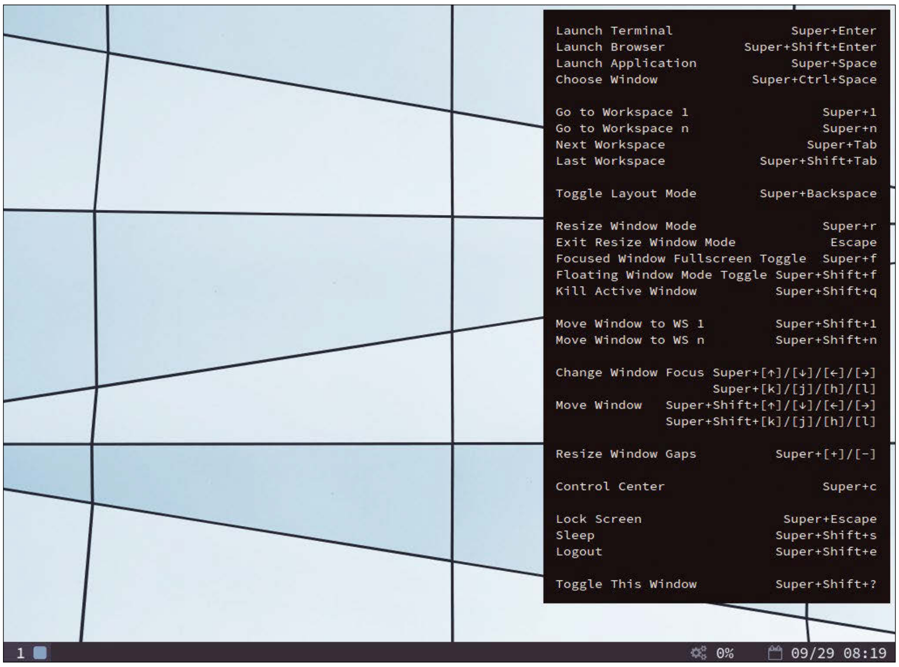
Listing 1: Converting Ubuntu

```
$ sudo add-apt-repository -y ppa:kgilmer/regolith-stable
$ sudo apt updatelib
$ sudo apt install regolith-desktop
```

You can call up the list of common keyboard shortcuts anytime by entering the key combination Super + Shift + ?. Until you have memorized all the shortcuts, the shortcut list is a fast and easy reference. Find your shortcut in the list and then enter Super + Shift + ? to toggle the list back off again.

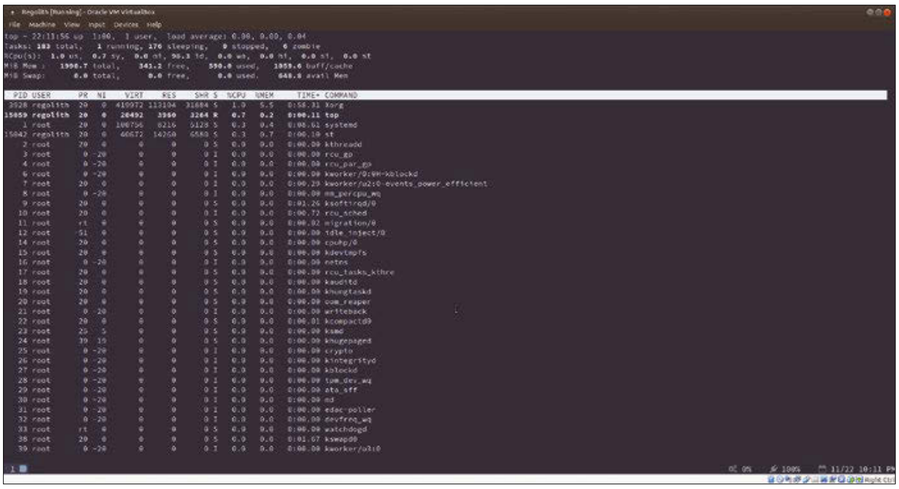
Users who don't like the mouse are often the same users who would rather work at the command line and avoid GUIs altogether. Regolith is well suited to the command-line user, but it is also fully capable of operating graphical applications within the structure of a tiling interface. In fact, one of the reasons Regolith exists is to support the full range of Ubuntu/Gnome applications within the i3 tiled window context. Keep in mind, however, that if you open a window with an application that requires a mouse, you'll still need to use the mouse and pointer within that window, unless you happen to know the keyboard shortcuts for the application. This context switching between a command-line and mouse environment might sound confusing, but once you get use to it, it is no different from switching between desktop apps and command-line tools in a conventional GUI environment. In fact, Regolith apps are mostly the same applications you would be using in everyday Ubuntu/Gnome – the only difference is the window navigation.

When you first launch Regolith, you'll get your first glimpse of the Regolith desktop (Figure 1). As you can see in the figure, the Regolith desktop looks quite a lot like other desktops – the most interesting feature might be the absence of a start menu,



```
Launch Terminal                     Super+Enter
Launch Browser                Super+Shift+Enter
Launch Application                  Super+Space
Choose Window                  Super+Ctrl+Space

Go to Workspace 1                       Super+1
Go to Workspace n                       Super+n
Next Workspace                        Super+Tab
Last Workspace                  Super+Shift+Tab

Toggle Layout Mode              Super+Backspace

Resize Window Mode                      Super+r
Exit Resize Window Mode                  Escape
Focused Window Fullscreen Toggle        Super+f
Floating Window Mode Toggle       Super+Shift+f
Kill Active Window                Super+Shift+q

Move Window to WS 1               Super+Shift+1
Move Window to WS n               Super+Shift+n

Change Window Focus Super+[↑]/[↓]/[←]/[→]
                    Super+[k]/[j]/[h]/[l]
Move Window    Super+Shift+[↑]/[↓]/[←]/[→]
               Super+Shift+[k]/[j]/[h]/[l]

Resize Window Gaps                Super+[+]/[-]

Control Center                          Super+c

Lock Screen                        Super+Escape
Sleep                             Super+Shift+s
Logout                            Super+Shift+e

Toggle This Window                Super+Shift+?
```

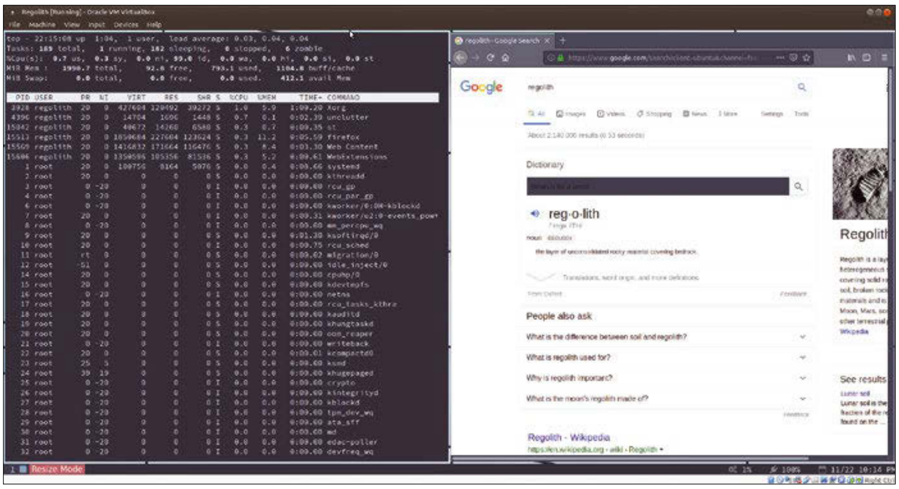
Figure 1: First view of Regolith – wallpaper plus a handy list of common shortcuts.



Regolith [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

```
top - 22:11:56 up  1:00,  1 user,  load average: 0.00, 0.00, 0.04
Tasks: 183 total,   1 running, 176 sleeping,   0 stopped,   6 zombie
%Cpu(s):  1.0 us,  0.7 sy,  0.0 ni, 98.3 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   1990.7 total,    341.2 free,    590.0 used,   1059.6 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.    648.8 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 2928 regolith  20   0  410972 113104  31684 S   1.0   5.5   0:58.31 Xorg
15059 regolith  20   0   20492   3960   3264 R   0.7   0.2   0:00.11 top
    1 root      20   0  100756   8216   5128 S   0.3   0.4   0:08.61 systemd
15042 regolith  20   0   40672  14260   6580 S   0.3   0.7   0:00.10 st
    2 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kthreadd
    3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
    6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-kblockd
    7 root      20   0       0      0      0 I   0.0   0.0   0:00.29 kworker/u2:0-events_power_efficient
    8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
    9 root      20   0       0      0      0 S   0.0   0.0   0:01.26 ksoftirqd/0
   10 root      20   0       0      0      0 I   0.0   0.0   0:00.72 rcu_sched
   11 root      rt   0       0      0      0 S   0.0   0.0   0:00.02 migration/0
   12 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/0
   14 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
   15 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kdevtmpfs
   16 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
   17 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_kthre
   18 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kauditd
   19 root      20   0       0      0      0 S   0.0   0.0   0:00.00 khungtaskd
   20 root      20   0       0      0      0 S   0.0   0.0   0:00.00 oom_reaper
   21 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 writeback
   22 root      20   0       0      0      0 S   0.0   0.0   0:00.01 kcompactd0
   23 root      25   5       0      0      0 S   0.0   0.0   0:00.00 ksmd
   24 root      39  19       0      0      0 S   0.0   0.0   0:00.00 khugepaged
   25 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 crypto
   26 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kintegrityd
   27 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kblockd
   28 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 tpm_dev_wq
   29 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 ata_sff
   30 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 md
   31 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 edac-poller
   32 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 devfreq_wq
   33 root      rt   0       0      0      0 S   0.0   0.0   0:00.00 watchdogd
   38 root      20   0       0      0      0 S   0.0   0.0   0:01.67 kswapd0
   39 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/u3:0
```

Figure 2: The first application fills the whole desktop and looks very much like any other Linux GUI.



Regolith [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

```
top - 22:15:08 up  1:04,  1 user,  load average: 0.03, 0.04, 0.04
Tasks: 189 total,   1 running, 182 sleeping,   0 stopped,   6 zombie
%Cpu(s):  0.7 us,  0.3 sy,  0.0 ni, 99.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   1990.7 total,     92.0 free,    793.1 used,   1104.8 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.    412.1 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 2928 regolith  20   0  427604 120492  39272 S   1.0   5.9   1:09.20 Xorg
 4396 regolith  20   0   14704   1696   1448 S   0.7   0.1   0:02.39 unclutter
15042 regolith  20   0   40672  14260   6580 S   0.3   0.7   0:00.35 st
15513 regolith  20   0 1850684 227604 123624 S   0.3  11.2   0:05.59 firefox
15569 regolith  20   0 1416832 171664 116476 S   0.3   8.4   0:03.30 Web Content
15606 regolith  20   0 1350596 105356  81536 S   0.3   5.2   0:00.61 WebExtensions
    1 root      20   0  100756   8164   5076 S   0.0   0.4   0:08.66 systemd
    2 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kthreadd
    3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
    6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-kblockd
    7 root      20   0       0      0      0 I   0.0   0.0   0:00.31 kworker/u2:0-events_powe
    8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
    9 root      20   0       0      0      0 S   0.0   0.0   0:01.30 ksoftirqd/0
   10 root      20   0       0      0      0 I   0.0   0.0   0:00.75 rcu_sched
   11 root      rt   0       0      0      0 S   0.0   0.0   0:00.02 migration/0
   12 root     -51   0       0      0      0 S   0.0   0.0   0:00.00 idle_inject/0
   14 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
   15 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kdevtmpfs
   16 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
   17 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_kthre
   18 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kauditd
   19 root      20   0       0      0      0 S   0.0   0.0   0:00.00 khungtaskd
   20 root      20   0       0      0      0 S   0.0   0.0   0:00.00 oom_reaper
   21 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 writeback
   22 root      20   0       0      0      0 S   0.0   0.0   0:00.01 kcompactd0
   23 root      25   5       0      0      0 S   0.0   0.0   0:00.00 ksmd
   24 root      39  19       0      0      0 S   0.0   0.0   0:00.00 khugepaged
   25 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 crypto
   26 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kintegrityd
   27 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kblockd
   28 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 tpm_dev_wq
   29 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 ata_sff
   30 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 md
   31 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 edac-poller
   32 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 devfreq_wq
```

Figure 3: The second application starts – the first one scoots over automatically, and the two windows share the space.

tool tray, or any other icon that would be a target for the user to point and click.

As you can see in Table 1, the web browser and terminal window applications are popular enough to get their own shortcuts. To launch a terminal window, press the Super + Enter key combination. The first application you launch fills the whole desktop (Figure 2). Interact with the terminal as you would on any other Linux system – you won't really notice anything different when you're working with a single application.

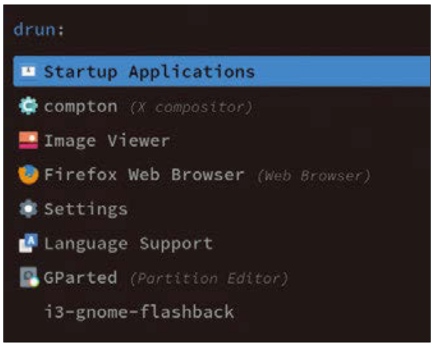
Suppose you want to search the web for more information on a command you plan to enter in the terminal. Press the Super + Shift + Enter combination to launch a browser window. The terminal moves over to the left automatically, and the browser launches on the right (Figure 3). In a single step, you have launched a second application and positioned both windows for full visibility. (In a typical WIMP-style system, the second application would launch on top of the first one, and you would then have to expend several steps moving windows around to achieve the perfect alignment shown in Figure 3.)

Suddenly you realize you'd better document what you're doing. Time to launch LibreOffice. By default, Regolith will continue to split the screen from left to right, adding new applications on the right side and sliding everything over. If you continue to add more applications, the windows will eventually get too narrow. Press the Super + Backspace key combination to toggle the Layout mode, which defines whether to add new windows vertically or horizontally.

Then enter Super + Space to launch an application menu based on the Rofi application launcher [9] (Figure 4). If LibreOffice Writer doesn't appear on the list, type it in the search field. The new application launches below the previously active window. If you think LibreOffice needs a little more screen real estate, and you can get by with less for the terminal, press Super + r to enter Resize Window mode, which will let you slide the borders between the windows using the arrow keys to achieve an optimum position (Figure 5). You'll need to press the Esc key to exit Resize Window mode before entering any additional commands.

Table 1: Regolith Keyboard Shortcuts

Sequence	Function
Applications and Commands	
Super+Enter	Launch terminal
Super+Shift+Enter	Launch Browser
Super+Space	Application launcher
Super+Shift+Space	Command launcher
Super+c	Open settings
Super+n	Open network configuration
Super+w	Open WiFi settings
Manipulating Windows	
Super+Ctrl+Space	Window selector
Super+Arrow keys	Navigate between windows
Super+Shift+Arrow keys	Reorganize windows
Super+Backspace	Toggle between vertical and horizontal mode
Super+f	(De)activate full-screen mode for windows
Super+Shift+f	Activate floating mode for current application
Super++, Super+-	Increasing or decreasing size of webs between segments
Super+Shift+?	Shows help for important keyboard shortcuts
Virtual Desktops	
Super+1 to 0	Changes to desktops 1 to 10
Super+Ctrl+1 to 9	Changes to desktops 11 to 19
Super+Alt+1 to 0	Moves current application to desktops 1 to 10
Super+Ctrl+Alt+1 to 9	Moves current application to desktops 11 to 19
Super+Tabulator	Changes to the next desktop
Super+Shift+Tab	Changes to the previous desktop
Super+t	Toggles between horizontal and vertical tiling and tab view
Super+r	Activates Resize mode (exit with Esc or Enter)
Manage Session	
Super+Shift+e	Ends session and logs out user
Super+Shift+s	Puts computer on standby
Super+Shift+b	Restarts computer
Super+Shift+p	Shuts down computer



```
drun:
Startup Applications
compton (X compositor)
Image Viewer
Firefox Web Browser (Web Browser)
Settings
Language Support
GParted (Partition Editor)
i3-gnome-flashback
```

Figure 4: Press Super+Space to access the application menu.

Change the active window (called the window *focus*) to the window on the left side, and then launch another application for a four-window configuration (Figure 6). Remember you can toggle to Resize Window mode to adjust the borders. To make a window active, click on it with the mouse, or move between the windows by pressing Super plus one of the arrow keys.

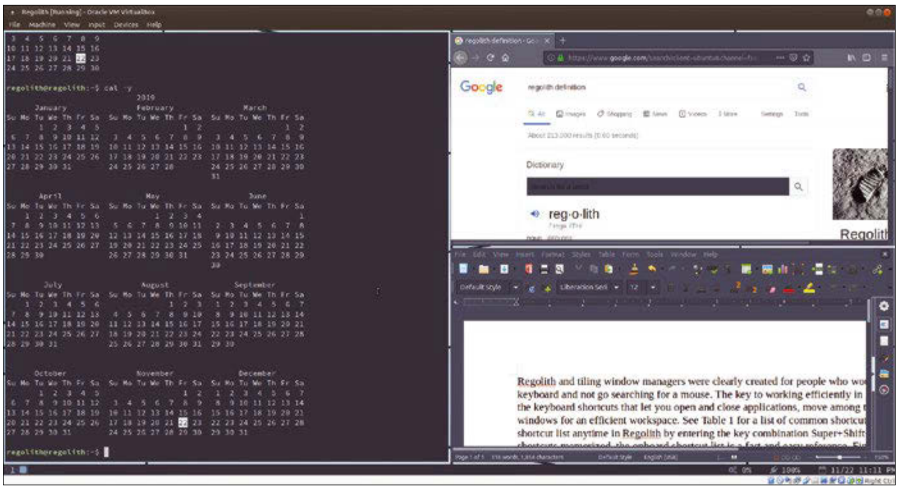
If you are mainly working in a single window and don't want the clutter of the whole tiled desktop, press Super + f to toggle the currently active window to full screen mode (which looks similar to Figure 2). To close the active window, press Super + Shift + q. Other key combinations change the workspace, lock the screen, or initiate the logout process.

Find Your Groove

The ideal screen configuration depends on the nature of your work and the applications you are using. A four-window desktop is fairly comfortable for most monitors. Command-line users often operate many more terminal windows at once – especially if all they need to see is the command line and a couple lines of output. Even in GUI scenarios, a user who is skilled with the key combinations can easily move among small windows and toggle them to full screen mode to work in one at a time.

If sliding around the window borders does not provide enough control of the layout, press Super + Shift + f to toggle to Floating Window mode, which lets you place the windows wherever you want them. Then press Super + Shift plus an arrow key to move the active window (or drag the window with the mouse).

If Regolith is installed on your computer (not operating as a live system), you can save the desktop configuration so it will be waiting for you the next time you log on. Arrange your desktop for efficient access to the four or five applications you use the most, and you'll gradually become accustomed to moving between the windows using the keyboard commands. (See the "Hacking" box.)

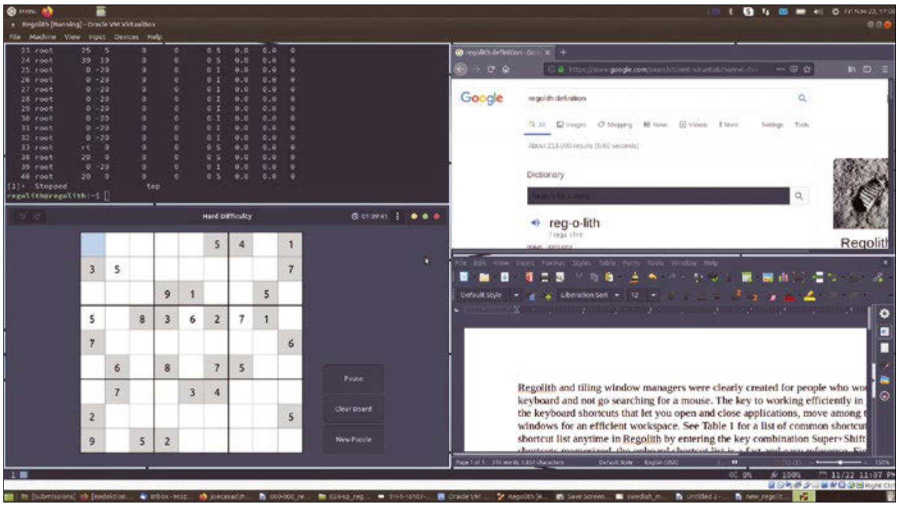


Regolith [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Google
regolith definition
Dictionary
reg·o·lith
Regolith
File Edit View Insert Format Styles Table Form Tools Window Help
Default Style
Liberation Seri
12

```
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30

regolith@regolith:~$ cal -y
                            2019
      January               February               March
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
       1  2  3  4  5                  1  2                  1  2
 6  7  8  9 10 11 12   3  4  5  6  7  8  9   3  4  5  6  7  8  9
13 14 15 16 17 18 19  10 11 12 13 14 15 16  10 11 12 13 14 15 16
20 21 22 23 24 25 26  17 18 19 20 21 22 23  17 18 19 20 21 22 23
27 28 29 30 31        24 25 26 27 28        24 25 26 27 28 29 30
                                            31

       April                  May                   June
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6            1  2  3  4                     1
 7  8  9 10 11 12 13   5  6  7  8  9 10 11   2  3  4  5  6  7  8
14 15 16 17 18 19 20  12 13 14 15 16 17 18   9 10 11 12 13 14 15
21 22 23 24 25 26 27  19 20 21 22 23 24 25  16 17 18 19 20 21 22
28 29 30              26 27 28 29 30 31     23 24 25 26 27 28 29
                                            30

        July                 August              September
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6               1  2  3   1  2  3  4  5  6  7
 7  8  9 10 11 12 13   4  5  6  7  8  9 10   8  9 10 11 12 13 14
14 15 16 17 18 19 20  11 12 13 14 15 16 17  15 16 17 18 19 20 21
21 22 23 24 25 26 27  18 19 20 21 22 23 24  22 23 24 25 26 27 28
28 29 30 31           25 26 27 28 29 30 31  29 30

      October               November              December
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
       1  2  3  4  5                  1  2   1  2  3  4  5  6  7
 6  7  8  9 10 11 12   3  4  5  6  7  8  9   8  9 10 11 12 13 14
13 14 15 16 17 18 19  10 11 12 13 14 15 16  15 16 17 18 19 20 21
20 21 22 23 24 25 26  17 18 19 20 21 22 23  22 23 24 25 26 27 28
27 28 29 30 31        24 25 26 27 28 29 30  29 30 31

regolith@regolith:~$
```

Regolith and tiling window managers were clearly created for people who wo
keyboard and not go searching for a mouse. The key to working efficiently in
the keyboard shortcuts that let you open and close applications, move among t
windows for an efficient workspace. See Table 1 for a list of common shortcut
shortcut list anytime in Regolith by entering the key combination Super+Shift

Figure 5: Toggle the Layout mode to launch a third application below. Resize Window mode lets you shift the window borders to give more space to applications that need it.



Regolith and tiling window managers were clearly created for people who wo
keyboard and not go searching for a mouse. The key to working efficiently in
the keyboard shortcuts that let you open and close applications, move among t
windows for an efficient workspace. See Table 1 for a list of common shortcut
shortcut list anytime in Regolith by entering the key combination Super+Shift

Figure 6: Four applications fit comfortably on most monitors, and many can manage more. If you need more space, enter Super+f to toggle the active window to full-screen mode.

Regolith also makes it easy to create different workspaces for different scenarios. For instance, workspace 1 could be the desktop you like to use at work, and workspace 2 could the configuration you like at home. You can tab between the workspaces by pressing Super + Tab or just press the Super key and type the number of the workspace you wish to reach.

Conclusions

It takes a while to become accustomed to a workflow designed for keyboard shortcuts, but once you get used to it, it is often easier to work with a tiling window manager than with a classic desktop environment. The benefits are particularly noticeable when you have to work with a small screen at low resolution, such as on a 13-inch business laptop. Since the Regolith desktop does without space-consuming panels or window ribbons, you have more room for content.

Regolith is a good candidate for introducing you to the world of tiling window managers. Even if you are used to relying on the mouse, you can master the basics of Regolith in just a few minutes. The project website provides a Getting Started guide [11] and other useful documents for new users. See the detailed HowTo list in the Regolith Wiki [12] for additional tips on customizing your system. ■■■

Listing 2: Editing the i3 Configuration

```
$ mkdir -p ~/.config/regolith/i3
$ cp /etc/regolith/i3/config ~/.config/regolith/i3/config
$ gedit ~/.config/regolith/i3/config
```

Info

[1] Xerox Star Professional Workstation: *https://www.youtube.com/watch?v=zVw86emu-K0*

[2] Gnome Material Shell: *https://github.com/PapyElGringo/material-shell*

[3] KDE Krohnkite: *https://github.com/esjeon/krohnkite*

[4] i3: *https://i3wm.org*

[5] Sway Window Manager: *https://swaywm.org/*

[6] Regolith: *https://regolith-linux.org*

[7] Download: *https://sourceforge.net/projects/regolith-linux/files/regolith-linux-r1.2*

[8] PPA package source: *https://launchpad.net/~kgilmer/+archive/ubuntu/regolith-stable*

[9] Rofi: *https://github.com/davatorium/rofi*

[10] Unixporn: *https://www.reddit.com/r/unixporn*

[11] Regolith Getting Started Guide: *https://github.com/regolith-linux/regolith-desktop/wiki/Getting-Started*

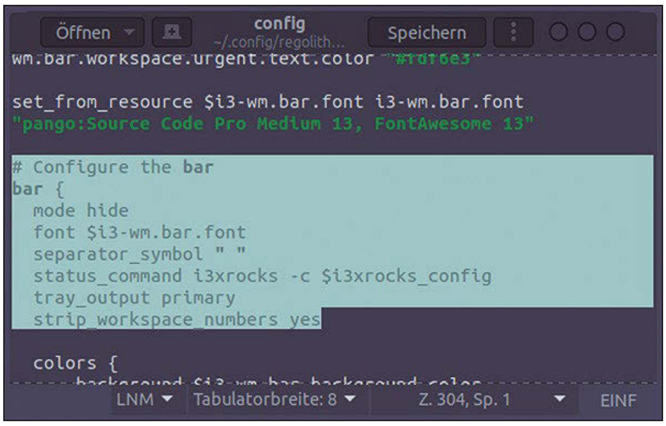
[12] Regolith HowTos: *https://github.com/search?utf8=%E2%9C%93&q=org%3Aregolith-linux+HowTo+in%3Atitle&type=Wikis*

Hacking

The configuration implemented by Regolith is just one example of how to design an i3 desktop. Users have countless other possibilities for adapting the system. If you need further inspiration, search the web for "Ricing i3" or take a look at Reddit Unixporn [10] – don't worry about the name, everything is legit on the site.

For example, Regolith does without a notification area, where active applications (like the network manager applet or a messenger like Skype) can log messages in the background. To enable a notification area, first create a copy of the configuration file responsible for i3 in the home directory and open the file in an editor (Listing 2).

Then change the `tray_output none` line to `tray_output primary` in the section following `# Configure the bar`. Optionally add a `mode hide` line below `bar {` – then the desktop bar at the bottom of the screen only appears when you press Super (Figure 7). In order to enable the new configuration file, you need to log off and back on again by pressing Super+Shift+E. For further changes, it is sufficient to instruct i3 to reread its configuration with Super+Shift+R.



```
wm.bar.workspace.urgent.text.color "#fdf6e3"

set_from_resource $i3-wm.bar.font i3-wm.bar.font
"pango:Source Code Pro Medium 13, FontAwesome 13"

# Configure the bar
bar {
  mode hide
  font $i3-wm.bar.font
  separator_symbol " "
  status_command i3xrocks -c $i3xrocks_config
  tray_output primary
  strip_workspace_numbers yes

  colors {
```

Figure 7: The i3 desktop is configured in a single file. The mode hide option causes the footer to appear only after pressing Super. The `tray_output primary` line enables a notification field in the bar for applications like the network manager applet or Skype.

DrupalCon

MINNEAPOLIS **2020**

MAY 18-22

Minneapolis Convention Center
events.drupal.org/minneapolis2020

Solidify your plans.
Register **today** to start your amazing DrupalCon journey!

More than 3,000 of the top digital minds will convene in Minneapolis to share knowledge, create solutions, build relationships, and shape the future.

Network, learn and be inspired.

Our session tracks provide something for **everyone**! The conference offers industry summits, trainings and content tailored to your expertise, including these specialties:

- CONTENT & DIGITAL MARKETING
- BUILDER
- AGENCY LEADERSHIP

A powerful IDE for Gnome

The Right Stuff

This integrated development environment for Gnome offers several features gedit lacks.

By Christoph Langner

Have you ever tried to knock a nail into a wall with a screwdriver or drill a hole with a saw? Of course not: You know to use the right tool. Then why would you use Gnome's gedit as your integrated development environment (IDE)?

When it comes to global searching, Git integration, and an integrated terminal, Gnome Builder is the right tool for programming in a Gnome environment. Gnome Builder emerged from a crowdfunding campaign [1] in 2015, but many developers still don't have it in their toolbox. Gnome Builder 3.34, released with Gnome 3.34, deserves its turn in the spotlight [2].

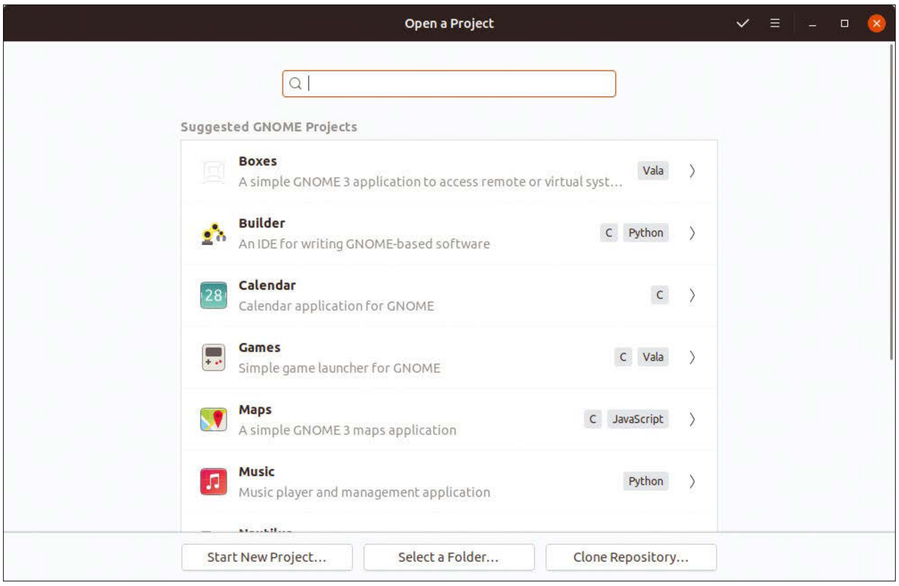
You can launch Gnome Builder directly from the application menu (see the "Installation" box); it starts up with a wizard that opens the last project you used or clones known Gnome applications like calendars, files, or photos (Figure 1). Alternatively you can start a new project by clicking on *Start New Project…* or clone any other project on the computer by selecting *Clone Repository…*. After importing, the system automatically offers to download the software development kits (SDKs) required to build your project.

Operation

In the sidebar to the left of the main window under Project Tree, you will find Version Control, Build Targets, Unit Tests, and Files. If you double-click on one of the source code files, Builder opens its contents as a page in the main window to the right (Figure 2). Pages are files opened in parallel by Builder. All of the currently active pages appear above the address tree under Open Pages. By clicking on one of the entries, using the left and right arrows next to the file name in the editor area, or pressing Alt + Ctrl + Page-Up/Page-Down, you can quickly toggle between the pages.

You can use the toolbar buttons top right to switch between Editor and Profiler mode (more on this later) or select *Build Preferences*, which takes you to a menu where you can create new files or open additional windows and terminals. The next two buttons to the right open and close the sidebar and footer. For example, Builder can output the build log in the lower area of the application window or display an integrated terminal window.

In the middle of the toolbar, Builder displays a field with the name of the current project; clicking on this field displays the project's details. Clicking on



Open a Project

Suggested GNOME Projects

Boxes – A simple GNOME 3 application to access remote or virtual syst… – Vala

Builder – An IDE for writing GNOME-based software – C, Python

Calendar – Calendar application for GNOME – C

Games – Simple game launcher for GNOME – C, Vala

Maps – A simple GNOME 3 maps application – C, JavaScript

Music – Music player and management application – Python

Figure 1: A wizard helps you create your first project. Code from existing projects can be cloned directly to your own computer using Builder.

Lead Image © Corina Rosu, 123rf.com

Installation

Every distribution maintained by Gnome in its package sources should also contain Gnome Builder. The installation for Debian, Ubuntu, or Linux Mint is handled directly with the graphical package manager or via:

```
sudo apt install gnome-builder
```

On Fedora, use:

```
sudo dnf install gnome-builder
```

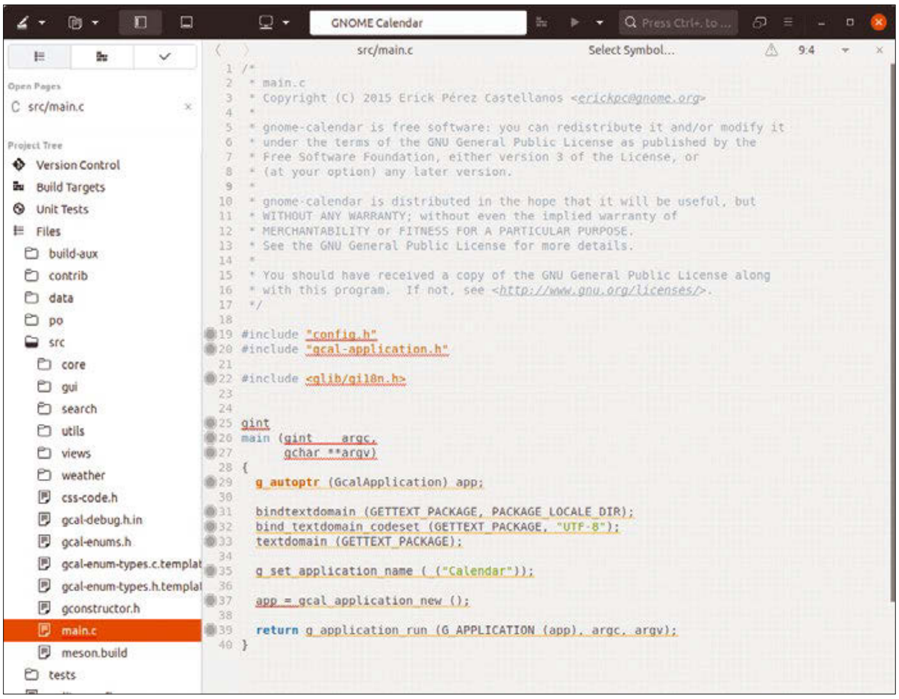
Arch Linux also includes Gnome Builder in its package sources.

However, the major distributions tend to lag behind when it comes to the Gnome Builder version. An update to the current version 3.34 is only available if you update the distribution, too. Ubuntu, for example, delivers Gnome 3.34 with Ubuntu 19.10 “Eoan Ermine.” However, at the time of writing this article, Gnome Builder 3.32 is the current version found in Eoan Ermine’s package sources.

Optionally, you can install Gnome Builder 3.34 independent of your distribution choice using a Flatpak, which you can pick up from Flathub [3]. On a test system with Ubuntu 19.04, the installation worked in this constellation without any complications. However, you need to prepare the system for installing Flatpaks following the instructions from the Flatpak website [4].

the brick wall icon lets you build the current project; clicking on the Play button next to it executes the project. Gnome Builder reports any errors or problems in the build output in the footer. The search field at the right edge of the toolbar not only searches in the currently active source code file, but also automatically returns matches from all of the project files. In the hamburger menu, you can start new projects or open the *Preferences*.

Gnome Builder covers an IDE’s most important functions. Builder becomes really interesting when you start to customize the workspace. Take a look at the bar at the top of the editor area, for example. Next to the arrows for toggling the page, you will find the name of the opened file, a jump function that automatically sends the cursor to a selected function, and the line number currently being edited. Below the icon with the down arrow, you’ll find a menu for configuring the workspace.



```
/*
 * main.c
 * Copyright (C) 2015 Erick Pérez Castellanos <erickpc@gnome.org>
 *
 * gnome-calendar is free software: you can redistribute it and/or modify it
 * under the terms of the GNU General Public License as published by the
 * Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * gnome-calendar is distributed in the hope that it will be useful, but
 * WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
 * See the GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License along
 * with this program.  If not, see <http://www.gnu.org/licenses/>.
 */

#include "config.h"
#include "gcal-application.h"

#include <glib/gi18n.h>


gint
main (gint    argc,
      gchar **argv)
{
  g_autoptr (GcalApplication) app;

  bindtextdomain (GETTEXT_PACKAGE, PACKAGE_LOCALE_DIR);
  bind_textdomain_codeset (GETTEXT_PACKAGE, "UTF-8");
  textdomain (GETTEXT_PACKAGE);

  g_set_application_name (_("Calendar"));

  app = gcal_application_new ();

  return g_application_run (G_APPLICATION (app), argc, argv);
}
```

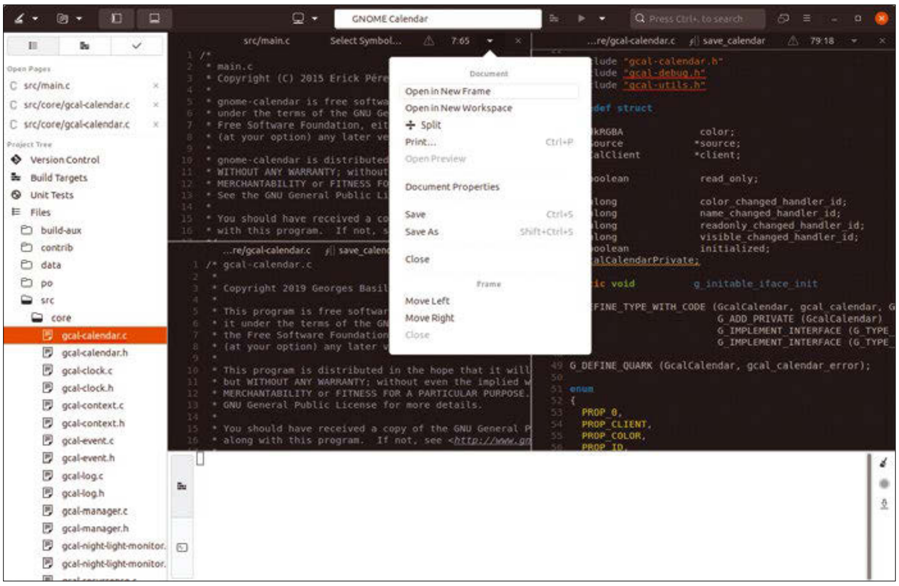
Figure 2: Builder 3.34 with one file opened. The sidebar containing the project tree can be hidden if necessary.

Flexible User Interface

The *Open in New Frame* option splits the workspace vertically into two (or more) sections. In the default setting, Builder displays the contents of the currently open page (that is, the file currently being edited) in parallel. Changes in one section automatically appear in the other. In addition, the *Split* option can be used to split the currently active frame horizontally (Figure 3).

Gnome Builder supports simultaneous editing of code or text at multiple locations in the currently opened file. Simply left-click on the desired positions while holding down Ctrl. Everything you then type will appear at the selected locations.

All the windows always display the same content. This means that you can make changes to several parts of a program file in parallel without having to scroll through the window. If you work

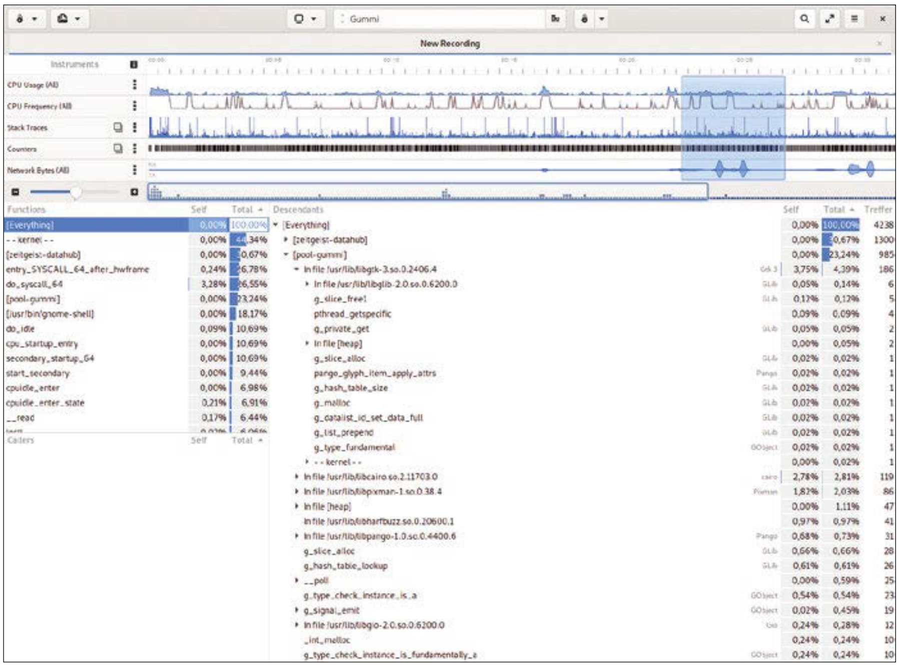


```
/*
 * main.c
 * Copyright (C) 2015 Erick Pére
 *
 * gnome-calendar is free softwa
 * under the terms of the GNU Ge
 * Free Software Foundation, eit
 * (at your option) any later ve
 *
 * gnome-calendar is distributed
 * WITHOUT ANY WARRANTY; without
 * MERCHANTABILITY or FITNESS FO
 * See the GNU General Public Li
 *
 * You should have received a co
 * with this program.  If not, s
```

```
/* gcal-calendar.c
 *
 * Copyright 2019 Georges Basil
 *
 * This program is free softwar
 * it under the terms of the GN
 * the Free Software Foundation
 * (at your option) any later v
 *
 * This program is distributed in the hope that it will
 * but WITHOUT ANY WARRANTY; without even the implied w
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General P
 * along with this program.  If not, see <http://www.gn
```

```
lude "gcal-calendar.h"
lude "gcal-debug.h"
lude "gcal-utils.h"

def struct

kRGBA              color;
ource             *source;
alClient          *client;

oolean             read_only;

long               color_changed_handler_id;
long               name_changed_handler_id;
long               readonly_changed_handler_id;
long               visible_changed_handler_id;
oolean             initialized;
alCalendarPrivate;

ic void          g_initable_iface_init

FINE_TYPE_WITH_CODE (GcalCalendar, gcal_calendar, G
                     G_ADD_PRIVATE (GcalCalendar)
                     G_IMPLEMENT_INTERFACE (G_TYPE_
                     G_IMPLEMENT_INTERFACE (G_TYPE_

G_DEFINE_QUARK (GcalCalendar, gcal_calendar_error);

enum
{
  PROP_0,
  PROP_CLIENT,
  PROP_COLOR,
  PROP_ID,
```

Figure 3: Several files can be displayed in the workspace in the form of frames. If required, Builder also lets you subdivide frames into several areas and thus edit different locations in a file simultaneously.



Functions	Self	Total
[Everything]	0,00%	100,00%
- - kernel - -	0,00%	44,34%
[zeitgeist-datahub]	0,00%	0,67%
entry_SYSCALL_64_after_hwframe	0,24%	26,78%
do_syscall_64	3,28%	26,55%
[pool-gummi]	0,00%	23,24%
[/usr/bin/gnome-shell]	0,00%	18,17%
do_idle	0,09%	10,69%
cpu_startup_entry	0,00%	10,69%
secondary_startup_64	0,00%	10,69%
start_secondary	0,00%	9,44%
cpuidle_enter	0,00%	6,98%
cpuidle_enter_state	0,21%	6,91%
__read	0,17%	6,44%

Callers	Self	Total

Descendants		Self	Total	Treffer
[Everything]		0,00%	100,00%	4238
[zeitgeist-datahub]		0,00%	0,67%	1300
[pool-gummi]		0,00%	23,24%	985
In file /usr/lib/libgtk-3.so.0.2406.4	Gtk 3	3,75%	4,39%	186
In file /usr/lib/libglib-2.0.so.0.6200.0	GLib	0,05%	0,14%	6
g_slice_free1	GLib	0,12%	0,12%	5
pthread_getspecific		0,09%	0,09%	4
g_private_get	GLib	0,05%	0,05%	2
In file [heap]		0,00%	0,05%	2
g_slice_alloc	GLib	0,02%	0,02%	1
pango_glyph_item_apply_attrs	Pango	0,02%	0,02%	1
g_hash_table_size	GLib	0,02%	0,02%	1
g_malloc	GLib	0,02%	0,02%	1
g_datalist_id_set_data_full	GLib	0,02%	0,02%	1
g_list_prepend	GLib	0,02%	0,02%	1
g_type_fundamental	GObject	0,02%	0,02%	1
- - kernel - -		0,00%	0,02%	1
In file /usr/lib/libcairo.so.2.11703.0	cairo	2,78%	2,81%	119
In file /usr/lib/libpixman-1.so.0.38.4	Pixman	1,82%	2,03%	86
In file [heap]		0,00%	1,11%	47
In file /usr/lib/libharfbuzz.so.0.20600.1		0,97%	0,97%	41
In file /usr/lib/libpango-1.0.so.0.4400.6	Pango	0,68%	0,73%	31
g_slice_alloc	GLib	0,66%	0,66%	28
g_hash_table_lookup	GLib	0,61%	0,61%	26
__poll		0,00%	0,59%	25
g_type_check_instance_is_a	GObject	0,54%	0,54%	23
g_signal_emit	GObject	0,02%	0,45%	19
In file /usr/lib/libgio-2.0.so.0.6200.0	Gio	0,24%	0,28%	12
_int_malloc		0,24%	0,24%	10
g_type_check_instance_is_fundamentally_a	GObject	0,24%	0,24%	10

Figure 4: Gnome Builder directly integrates Sysprof, which helps programmers find bottlenecks and memory hogs in the program code.

with multiple monitors, you can use the *Open in New Workspace* option to open another code window and move it onto the second screen.

To help optimize your program's performance or find errors, Builder integrates Sysprof, a system-wide profiler for Linux [5]. To do this, instead of starting your program directly via the Play button, use the drop-down menu next to the button and select the *Run with Profiler* option.

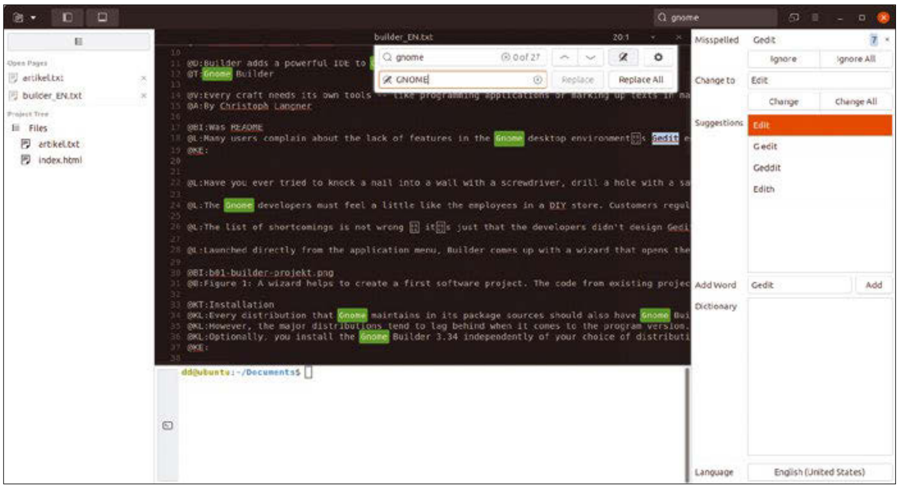
When you do this, Builder requests administrative rights and then logs the system's activities until you either close the application or click *Stop Recording* in the Builder editor area. The program then automatically opens the recording, from which you can then read the CPU history, the generated network traffic, or stack traces and counters over time (Figure 4).

Editing Text

You don't have to be a Gnome developer to use Builder. For people who want to work on plain old text or Markdown files, the program includes a powerful search function, as well as an integrated spellchecker. You can start searching by pressing Ctrl + F. Pressing the button next to the up and down arrows expands the search dialog and gives you a *Replace* option. The button on the far right adds options for regular expressions, uppercase and lowercase, and restricting the search to whole words. Pressing the Esc button closes the dialog again.

The spellchecker can be accessed by clicking in the text and selecting the spellchecker option in the context menu. As is usual with other text editors, Builder will immediately highlight misspelled words or expressions that are not in the dictionary with a red underscore. It also displays a sidebar on the right side of the screen, where you can go through the text word by word. You can correct or ignore each unknown word individually, with Builder usually providing helpful suggestions (Figure 5).

Optionally, you can select *Highlighting* | *Spellchecking* in the context menu of the editor window to easily highlight spelling mistakes.



```
@D:Builder adds a powerful IDE to 
@T:Gnome Builder

@V:Every craft needs its own tools -- like programming applications or marking up texts in ma
@A:By Christoph Langner

@BI:Was README
@L:Many users complain about the lack of features in the Gnome desktop environment's Gedit e
@KE:


@L:Have you ever tried to knock a nail into a wall with a screwdriver, drill a hole with a sa

@L:The Gnome developers must feel a little like the employees in a DIY store. Customers regul

@L:The list of shortcomings is not wrong – it's just that the developers didn't design Gedi

@L:Launched directly from the application menu, Builder comes up with a wizard that opens the

@BI:b01-builder-projekt.png
@B:Figure 1: A wizard helps to create a first software project. The code from existing projec

@KT:Installation
@KL:Every distribution that Gnome maintains in its package sources should also have Gnome Bui
@KL:However, the major distributions tend to lag behind when it comes to the program version.
@KL:Optionally, you install the Gnome Builder 3.34 independently of your choice of distributi
@KE:

dd@ubuntu:~/Documents$ 
```

Figure 5: Not just for programmers: Builder functions as an editor for simple text files or Markdown documents (e.g., this article was originally written in Builder).

Conclusions

Gnome is notorious for occasionally removing features that many users find useful. Gedit has an (unofficial) extension that allows the program's editor area to be divided into two parts. However, this no longer works in the current version of the Gnome desktop. Consequently, Builder's features come in handy here: The program can do everything that gedit can't.

Additionally, Builder's *Preferences* allow you to customize the IDE in detail – from a dark theme, through color schemes, fonts, line spacing, line breaks, to the scrolling behavior. Builder can also emulate keystrokes in the style of Vim, Emacs, or Sublime Text. Personally, I use Gnome Builder to write and edit articles for *Linux Magazine*. ■■■

Info

[1] Gnome Builder crowdfunding campaign: *https://www.indiegogo.com/projects/builder-an-ide-of-our-gnome#/*

[2] Gnome Builder: *https://wiki.gnome.org/Apps/Builder*

[3] Builder on Flathub: *https://flathub.org/apps/details/org.gnome.Builder*

[4] Setting up Flatpak: *https://flatpak.org/setup*

[5] Sysprof: *http://www.sysprof.com*

Simplifying file management with GNU Stow

Safely Stowed Away

Stow helps you manage custom configuration files across multiple users and computers, as well as organize your dotfiles. *By Ferdinand Thommes*

When it comes to file management, keeping chaos in check is highly recommended. Stow [1], a lesser-known GNU file management tool, functions as a symlink farm manager that organizes various sets of software or data located in separate directories on the filesystem, making them appear to be installed in a single directory tree.

Stow was originally created to manage software compiled on the system, which typically ends up in the subdirectories of `/usr/local/`. Stow is often used to install software globally across the system, but it also provides a clear-cut mechanism for managing configuration files and scripts in these installation's home directories. Stow can be considered the GNU project's official solution to keeping custom configuration files separate and distributing them to other systems.

When it comes to managing custom-compiled software, the software often does not come with an `uninstall` target in the makefile or a similar mechanism to get rid of it when needed. As a result, this can lead to a time-consuming manual uninstall. For instance, you would need to use `whereis <software>` to find out which subdirectories the software in question has been distributed to and then remove the files in each directory with `rm -rf`. Stow can simplify this process.

Listing 1: Creating stow/

```
$ ./configure --prefix=/usr/local/stow/<software>
$ make
# make install
```

Listing 2: Installing curl-7.66.0

```
$ ./configure --prefix=/usr/local/stow/curl-7.66.0
$ make
# make install
```

Three Steps and a Prefix

Stow creates a separate directory named `stow/` below `/usr/local/` during the installation. You specify this when compiling an application in the first step of the typical three-step installation of the software below `/usr/local/stow/`. Listing 1 shows you what the command looks like.

Often CMake is used in the first step. With `--prefix` as an additional instruction, all the software now ends up in `stow/<software>.<version>`. To convert it to the directory structure provided for by the Filesystem Hierarchy Standard (FHS) [2], type

```
cd /usr/local/stow/
```

to change to the `stow` directory, and trigger the creation of the symlinks to `/usr/local/` with `stow <software>`.

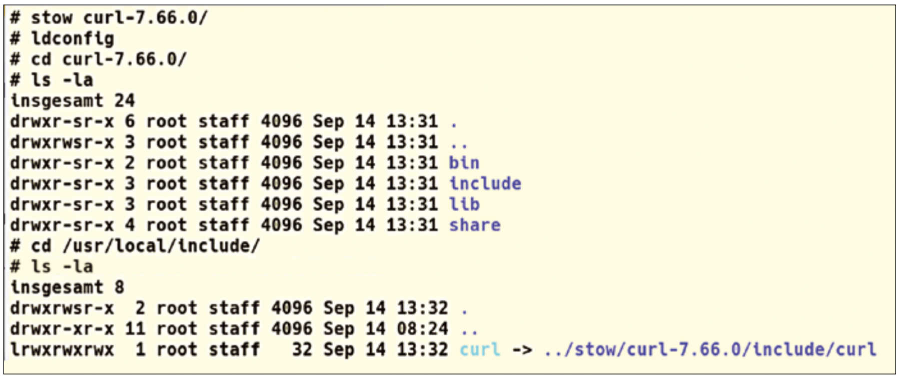
If you want to completely remove the software later on, all you need to do is type

```
stow -D <Software>
```

followed by

```
rm -rf <software>.<version>
```

Lead Image © zentilia, 123RF.com



```
# stow curl-7.66.0/
# ldconfig
# cd curl-7.66.0/
# ls -la
insgesamt 24
drwxr-sr-x 6 root staff 4096 Sep 14 13:31 .
drwxrwsr-x 3 root staff 4096 Sep 14 13:31 ..
drwxr-sr-x 2 root staff 4096 Sep 14 13:31 bin
drwxr-sr-x 3 root staff 4096 Sep 14 13:31 include
drwxr-sr-x 3 root staff 4096 Sep 14 13:31 lib
drwxr-sr-x 4 root staff 4096 Sep 14 13:31 share
# cd /usr/local/include/
# ls -la
insgesamt 8
drwxrwsr-x  2 root staff 4096 Sep 14 13:32 .
drwxr-xr-x 11 root staff 4096 Sep 14 08:24 ..
lrwxrwxrwx  1 root staff   32 Sep 14 13:32 curl -> ../stow/curl-7.66.0/include/curl
```

Figure 1: By installing the application in its own directory, Stow isolates the data from the system. It then adds symlinks in the directories to which the software is usually distributed.

After a package update, you can update the symlinks with `--restow`.

In my lab, I used the sources of the recently published cURL 7.66.0 as an example. The software is installed using the commands from Listing 2. After this, the `/usr/local/stow/curl-7.66.0` directories are located in the `bin/`, `include/`, `lib/`, and `share/` directories. If the prefix had not specified `../stow/curl-7.66.0/` as the installation location, the makefile would have distributed the program to these folders.

Symlink Redirection

To spoof the locations the system expects (Figure 1), you need to create symlinks from the `/usr/local/stow/` directory with:

```
stow curl-7.66.0
```

Follow this up with `ldconfig` to register new libraries with the system. If you now run `ls -la` in `/usr/local/include`, you will see the symlink to:

```
curl -> ../stow/curl-7.66.0/include/curl
```

The same applies to the other directories.

By installing software to a single directory and achieving FHS compliance through symlinks to the system, Stow resembles distributions such as NixOS or GoboLinux, which have long used a similar approach as a general solution for managing software.

Managing Dotfiles

Today, software is mainly installed using the package manager or it enters the system in the form of Flatpaks and Snaps. Over time, Stow has extended its application beyond its original purpose. This is reflected in the way the helper is implemented as a Perl script. In 2013, Stow's man page called the tool a "Software Package Installation Manager," while today it describes itself as a "Symlink Farm Manager" that offers extended application options for private use.

One extended application option includes managing dotfiles (the hidden configuration files in an installation's home directory called dotfiles because of the dot in front of their name). If you want to use these files or a collection of scripts that you use repeatedly on several computers, Stow can help.

One good example of this is the `.bashrc` file. This configuration file goes into action every time you open a terminal. Among other things, it contains useful entries, such as the aliases you created for frequently used command sequences or instructions for a custom prompt display. My `.bashrc` file has been in use for well over a decade, and I include it in every new installation.

Quick Setup

If you need to install software on multiple computers or test distributions on virtual machines, you can save a huge amount of time if applications reliably exhibit the same behavior. The overhead involved in setting up Stow for this purpose is manageable, because with Stow you only manage the dotfiles that you consider useful for this purpose.

First, for convenience, create a `dotfiles/` folder in your home directory. In the folder, create directories for all the programs whose dotfiles you want to manage with Stow. Figure 2 shows a sample home directory structure.

According to the XDG Base Directory specification [3], configuration files should be located in `$HOME/.config/`. However, not all developers stick to this, so the files are distributed all over the home directory. Mozilla, among others, sets a bad example here.

Tidying Up

Looking at the `.bashrc` file example again, the file is usually located directly in the home directory. To manage it with Stow, move it to the `dotfiles/` folder you previously created in the home directory. If you also want to manage the `.bash_profile` and `.bash-logout` files there, it makes sense to create a `bash/` folder in `dotfiles/` and move the three files there for better organization.

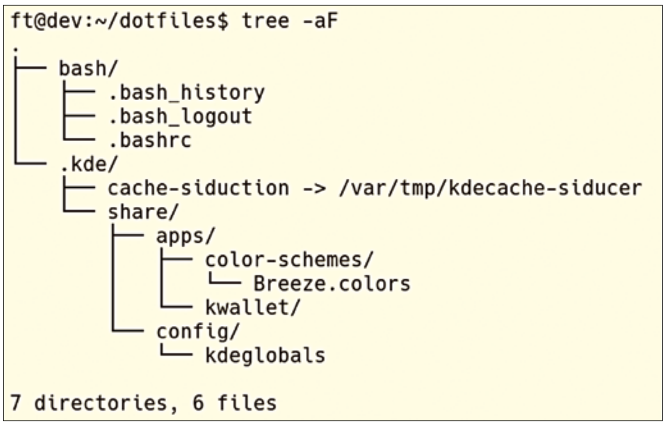
Then execute the command

```
cd ~/dotfiles && stow -v bash
```

to create the required symlinks (Figure 3). To see whether this worked, type

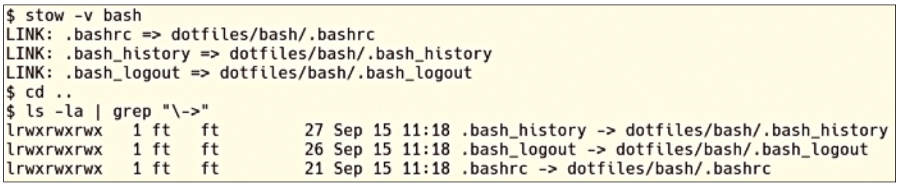
```
ls -la | grep "\->"
```

The command should show you the three symbolic links created by Stow. Using `stow *` lets you work on multiple directories in a single action.



```
ft@dev:~/dotfiles$ tree -aF
.
├── bash/
│   ├── .bash_history
│   ├── .bash_logout
│   └── .bashrc
└── .kde/
    ├── cache-siduction -> /var/tmp/kdecache-siducer
    └── share/
        ├── apps/
        │   ├── color-schemes/
        │   │   └── Breeze.colors
        │   └── kwallet/
        └── config/
            └── kdeglobals

7 directories, 6 files
```

Figure 2: The `tree -a` command lets you view the directory hierarchy as a tree structure. You then can check whether the replicated structure actually matches the conditions in your home directory.



```
$ stow -v bash
LINK: .bashrc => dotfiles/bash/.bashrc
LINK: .bash_history => dotfiles/bash/.bash_history
LINK: .bash_logout => dotfiles/bash/.bash_logout
$ cd ..
$ ls -la | grep "\->"
lrwxrwxrwx   1 ft   ft        27 Sep 15 11:18 .bash_history -> dotfiles/bash/.bash_history
lrwxrwxrwx   1 ft   ft        26 Sep 15 11:18 .bash_logout -> dotfiles/bash/.bash_logout
lrwxrwxrwx   1 ft   ft        21 Sep 15 11:18 .bashrc -> dotfiles/bash/.bashrc
```

Figure 3: The actual work is done by the `stow` command, followed by the application to be processed. Adding `--verbose` or `-v` shows you the symlinks created by this step. To check for their presence in the right location, run the `ls` command.

Stow creates links to the parent directory if not told to do something else (i.e., it typically uses its own home directory). If you want to link to another user's home directory, add the `--target` (`-t`) parameter. After changes to configuration files, use the `--restow` or `-R` parameter to reload them.

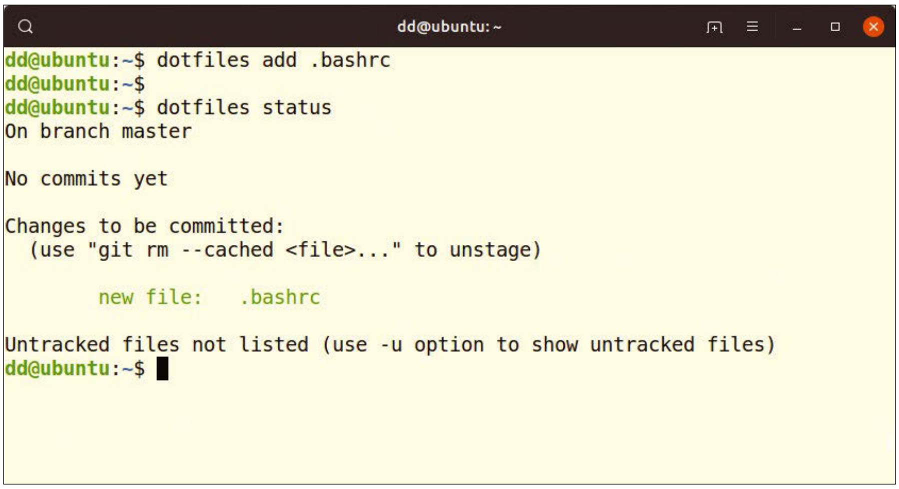
Freedom

If you enjoy working at the command line, you may have created several profiles for your preferred terminal emulation that define the font, its size, and the background color. For example, I maintain some Konsole profiles along with one Yakuake profile.

The files are located in `.config/` and `.config/share/` in the case of the Konsole. You therefore need to replicate the structure in `dotfiles/`. Since there are several files to back up here, it makes sense to create a subfolder named `konsole/` first and emulate the directory structure in it.

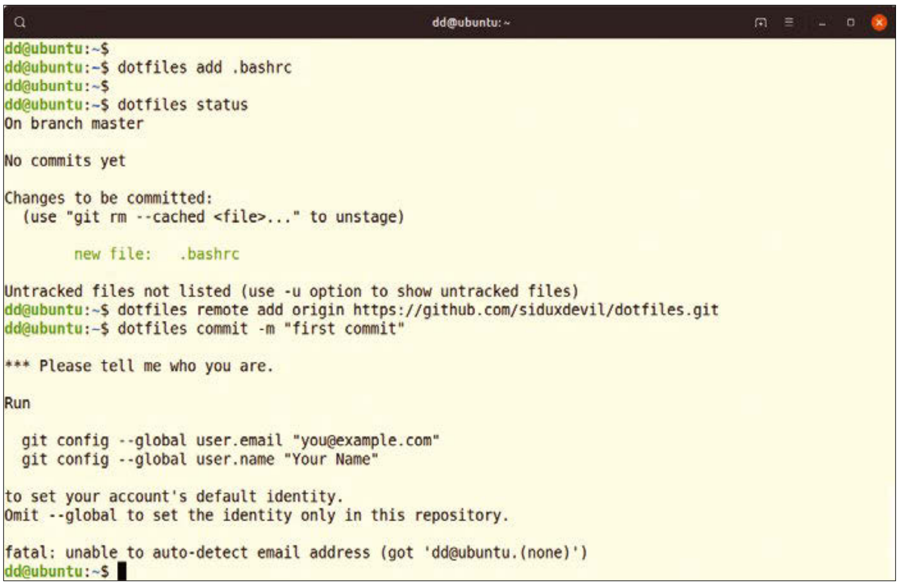
You can extend the directory structure as you like, for example, to store files containing private information separately or to sort files by subject. No matter how nested the structure is, as long as you run the `stow` command from the correct directory at the end of the day, the tool will do the rest. If you treat all dotfiles to be backed up in this way, you will always have fast access to them.

Another possibility is to manage dotfiles with Git. This can be done manually in a few minutes or automatically using a script such as yadm (Yet Another Dotfile Manager) [4] (Figure 4). This also makes it possible to encrypt directories that are under version control, but that you do not want to be openly accessible on GitHub. GitHub dedicates a separate page to the dotfiles topic [5].



```
dd@ubuntu:~$ dotfiles add .bashrc
dd@ubuntu:~$
dd@ubuntu:~$ dotfiles status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   .bashrc

Untracked files not listed (use -u option to show untracked files)
dd@ubuntu:~$
```

Figure 4: If you want to manage dotfiles with Git, first create a local repository and add the files to be managed to it.



```
dd@ubuntu:~$
dd@ubuntu:~$ dotfiles add .bashrc
dd@ubuntu:~$
dd@ubuntu:~$ dotfiles status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   .bashrc

Untracked files not listed (use -u option to show untracked files)
dd@ubuntu:~$ dotfiles remote add origin https://github.com/siduxdevil/dotfiles.git
dd@ubuntu:~$ dotfiles commit -m "first commit"

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'dd@ubuntu.(none)')
dd@ubuntu:~$
```

Figure 5: A repository on GitHub, or with your trusted provider, is connected to the local repository. It then versions private or public changes to the dotfiles.

Conclusions

Following the Unix philosophy, GNU Stow specializes in simplifying file management tasks. It is very useful if you want to distribute dotfiles locally between users or computers. If it is a question of making dotfiles available to other users beyond your own network, it is a good idea to extend Stow to include the `home` script from GitHub [6] (Figure 5). Stow's documentation [7] is, as with most GNU projects, very detailed. The tool does exactly what it says on the box with minimal effort. I did not experience a single error in my test. ■■■

Info

[1] Stow: *https://www.gnu.org/software/stow/*

[2] FHS: *https://en.wikipedia.org/wiki/Filesystem_Hierarchy_Standard*

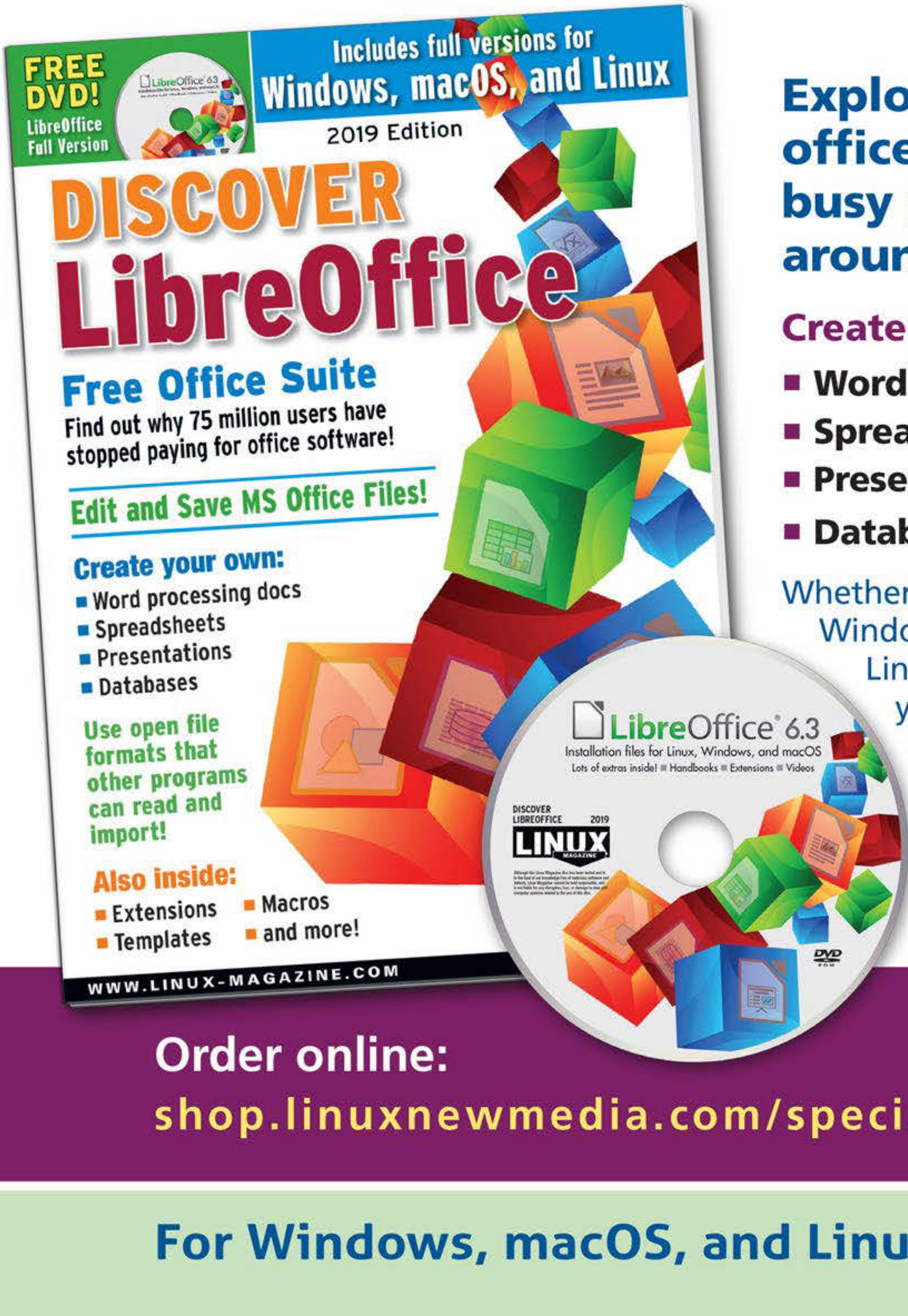
[3] XDG: *https://wiki.archlinux.org/index.php/XDG_Base_Directory*

[4] yadm: *https://yadm.io/*

[5] GitHub notes on dotfiles: *http://dotfiles.github.io/*

[6] home: *https://github.com/cdwilson/home*

[7] Stow documentation: *https://www.gnu.org/software/stow/manual/stow.html*


FREE DVD!
LibreOffice Full Version
Includes full versions for Windows, macOS, and Linux
2019 Edition
DISCOVER LibreOffice
Free Office Suite
Find out why 75 million users have stopped paying for office software!
Edit and Save MS Office Files!
Create your own:
Word processing docs
Spreadsheets
Presentations
Databases
Use open file formats that other programs can read and import!
Also inside:
Extensions
Templates
Macros
and more!
WWW.LINUX-MAGAZINE.COM
LibreOffice 6.3
Installation files for Linux, Windows, and macOS
Lots of extras inside! Handbooks Extensions Videos
DISCOVER LIBREOFFICE 2019
LINUX MAGAZINE
DVD

Store data securely in the cloud with Cryptomator

Secure Cloud

Cloud services often place little value on data encryption. With Cryptomator, you can easily and transparently encrypt your data locally before uploading to the cloud.

By Erik Bärwaldt and Thomas Leichtenstern

Cloud services provide a convenient and cost-effective alternative to local storage, especially for users who want to access their data from anywhere. However, many cloud providers do not pay sufficient attention to data security. They often store unencrypted data in the cloud as well as transferring the data without encryption. This paves the way for hackers to sniff for authentication data to access a cloud account and then spy on the data.

State institutions, such as intelligence services or investigative authorities, can also view this unprotected data. In addition, this kind of sniffing is often legal, especially in countries where data protection is of little importance. Cryptomator [1], a program developed by the German company Skymatic, puts a stop to this data espionage by encrypting your data transparently. Since Cryptomator's source code is licensed under the GPLv3, built-in backdoors are eliminated.

Strategy

Cryptomator works as a local server that processes the data to be encrypted on a virtual drive integrated by the Filesystem in Use (FUSE) module on Linux. If FUSE is not available, the software uses WebDAV instead. Cryptomator always encrypts the data with a 256-bit AES key and a MAC master key, generating the keys using scrypt technology [2]. In contrast to many other cryptographic programs, Cryptomator not only encrypts the file contents, but also their metadata. In addition, it changes the file size, which makes it difficult to draw conclusions about a file's content.

The software has an easy-to-use graphical front end that works with vaults where you store the data. The vaults match the directories to be synchronized with the cloud service; in other words, the front end must be connected to some kind of cloud storage. The files can be edited at will in the respective vault; encryption and decryption takes place practically in real time. The cloud service client then transfers the locally encrypted data to the server without transferring the keys.

Installation

Cryptomator provides the software for Linux as an AppImage, which means that it can be used on all current distributions without needing to retrofit any dependencies. For Ubuntu and its derivatives as of version 18.04 and for Arch Linux, the project provides separate repositories. For all other distributions, you first need to download the AppImage [3], which weighs in at around 55MB. The manufacturer asks for a donation of up to EUR25, but even if you choose not to donate, you can still download the application.

Note that Cryptomator only runs on computer systems with a 64-bit architecture. After downloading, assign execute permissions to the image. Then run it with the

```
./cryptomator-1.4.15-x86_64.AppImage
```

or just by clicking if you are using a file browser like Dolphin.

Getting Started

After starting the software, a two-part window without a menubar or buttonbar appears. The application is controlled using the three buttons located bottom left. Use the plus button to create

Lead Image © alexmillos, 123RF.com

new vaults, the minus button to remove existing ones from the interface, and the gear button to open a simple settings dialog, where you can define whether the software automatically checks for updates at startup and which function it uses to mount drives (Figure 1).

A click on the plus button opens a context menu in which you select + *Create vault*. Cryptomator displays the dialog for creating a new vault in an overlapping window, where you first define the vault's name and the directory in which the data are stored. Its name needs to match that of a cloud file directory. After pressing *Save*, the window closes. You are then prompted for a password for the vault in the main window. The color bar below indicates the password's strength.

After confirming the password by clicking on *Create Vault*, you will see the vault name and path in the left column. There is a closed padlock icon to the left. On the right, enter the password for the vault again and click on *Unlock Vault*.

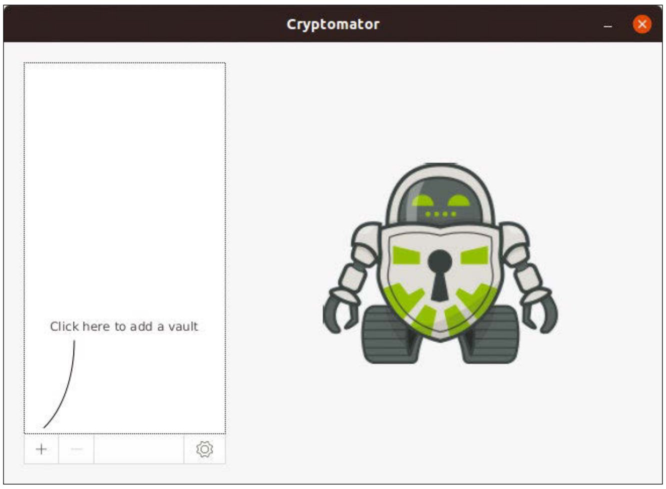


Figure 1: The Cryptomator front end is limited to the bare essentials and is easy to use.

Pressing the *More options* button takes you to further settings. You can enter your own drive options for mounting the device in an input field or tell the software to mount as *read-only*.

Opening and Closing

After unlocking the vault, Cryptomator opens a window with a file manager containing the virtual drive. At the same time, the padlock symbol in front of the relevant vault opens. You can quickly see which vaults are open, even if there are several (Figure 2).

You can drag and drop to move, or simply copy, the files to be encrypted to the virtual drive. In the main window, you will also see a graph in the right window segment that shows the transfer rates for encryption and decryption in megabytes per second. The red line indicates the encryption throughput, while the green line depicts decryption throughput (Figure 3). The software also opens several vaults simultaneously if required. In our lab, Cryptomator showed no weaknesses, even when encrypting and decrypting files with a size of more than 15GB.

As soon as changes (due to users storing and encrypting new files) occur on the virtual Cryptomator drive, the cloud client synchronizes the data with the server. This ensures that only encrypted data are sent to the cloud server. The files can be viewed and downloaded in the web interface. However, Linux identifies them as binary data; the content is not revealed either by name or file size. To stop working with Cryptomator, click *Lock Vault* on the right side of the main window. The software then closes the vault, which is indicated by the locked padlock icon.

When you open a vault again, you can immediately work with the decrypted files on the virtual drive. When loading and saving, the software then displays the data rates again in the graphical interface. If you close Cryptomator without closing the open vaults, the window is minimized to the taskbar, where you can access it at any time.

To remove a vault, click the minus icon in the lower left corner of the main window. If the vault is open, it is grayed, and the action cannot be performed. If the vault is closed, a window appears to inform you that the software is not removing the vault (the actual directory containing the data) from the system, but only the link from the graphical front end. You therefore need to manually delete the target directory from the system afterwards (Figure 4).

Resource Hog

We noticed some unusual behavior when running Cryptomator in our lab. For inexplicable reasons, the processor

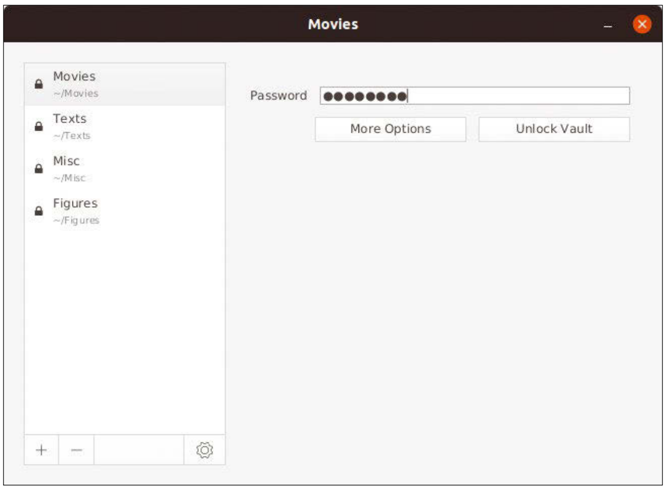


Figure 2: Cryptomator also manages multiple vaults simultaneously.

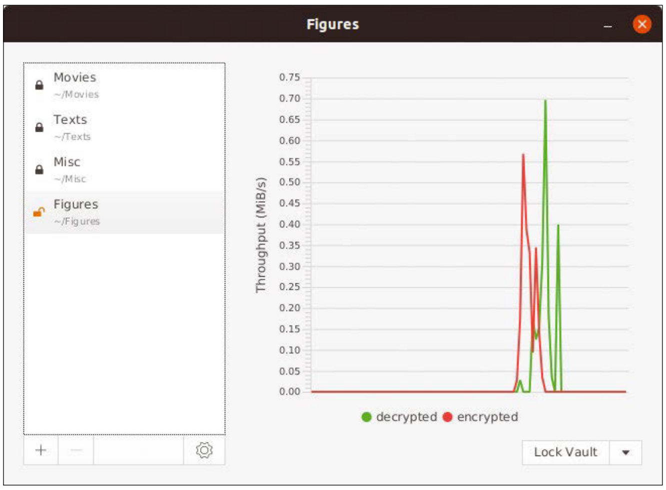
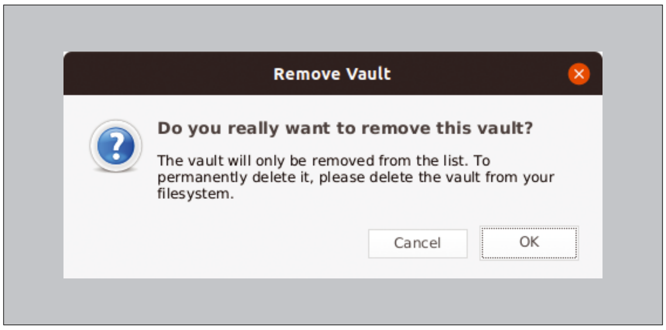


Figure 3: Cryptomator visualizes the transfer rates during encryption and decryption.

load and memory consumption of the system increased until the computer almost came to a standstill.

On launching Cryptomator, the resource consumption was normal. The same was true of mounting a vault and for as long as the vault remained open. But after closing a vault, the load on a multicore CPU increased to over 50 percent on all cores. The memory consumption increased continuously until the system started to swap out and became virtually impossible to use. We noticed that Cryptomator was eating up about 2GB RAM per minute (Figure 5). This started as soon as the application prompted us for the password for a suspended vault. Closing the software released the resources immediately.

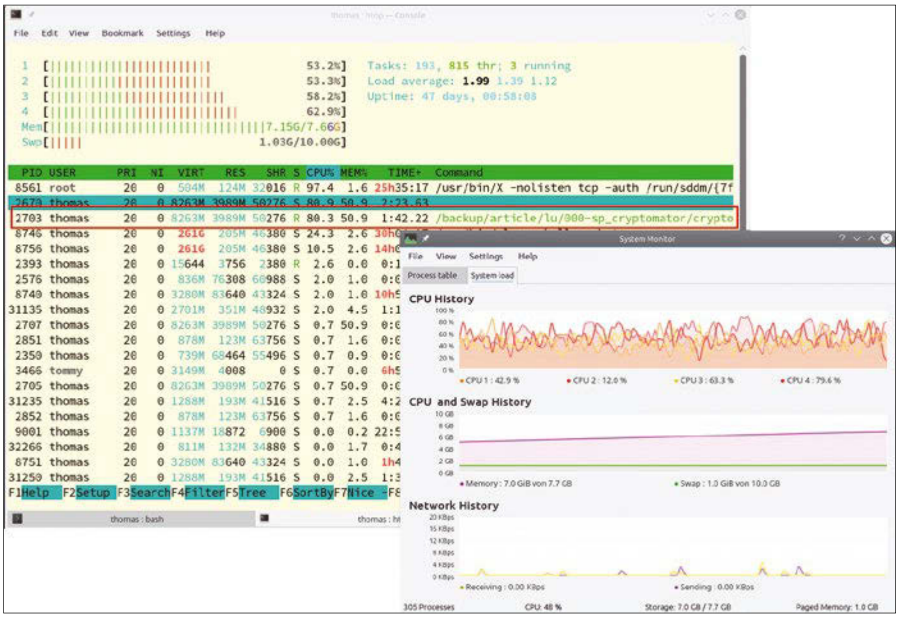
Further research revealed the following: This peculiarity occurs if you use the `cryptomator-1.4.15` AppImage offered by the project on openSUSE. On Ubuntu 19.04, the processor load increases permanently to 100 percent, but without depleting the RAM. This is primarily due to the `gnome-shell` process, which immediately returns to its original value of one to two percent CPU load after closing the AppImage.


Remove Vault

Do you really want to remove this vault?

The vault will only be removed from the list. To permanently delete it, please delete the vault from your filesystem.

Figure 4: The software only removes the entries for vaults; the data contained in the vaults are retained.



```
  1  [||||||||||||||||||||||||          53.2%]   Tasks: 193, 815 thr; 3 running
  2  [||||||||||||||||||||||||          53.3%]   Load average: 1.99 1.39 1.12
  3  [||||||||||||||||||||||||||        58.2%]   Uptime: 47 days, 00:58:08
  4  [||||||||||||||||||||||||||||      62.9%]
  Mem[|||||||||||||||||||||||||||||||7.15G/7.66G]
  Swp[|||||                    1.03G/10.00G]

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
 8561 root       20   0  584M  124M 32016 R 97.4  1.6 25h35:17 /usr/bin/X -nolisten tcp -auth /run/sddm/{7f
 2679 thomas     20   0 8263M 3989M 50276 S 80.9 50.9  2:23.63
 2703 thomas     20   0 8263M 3989M 50276 R 80.3 50.9  1:42.22 /backup/article/lu/000-sp_cryptomator/crypto
 8746 thomas     20   0  2616  205M 46380 S 24.3  2.6 30h
 8756 thomas     20   0  2616  205M 46380 S 10.5  2.6 14h
 2393 thomas     20   0 15644  3756  2380 R  2.6  0.0  0:1
 2576 thomas     20   0  836M 76308 60988 S  2.0  1.0  0:0
 8740 thomas     20   0 3280M 83640 43324 S  2.0  1.0 10h5
31135 thomas     20   0 2701M  351M 48932 S  2.0  4.5  1:1
 2707 thomas     20   0 8263M 3989M 50276 S  0.7 50.9  0:0
 2851 thomas     20   0  878M  123M 63756 S  0.7  1.6  0:0
 2350 thomas     20   0  739M 68464 55496 S  0.7  0.9  0:0
 3466 tommy      20   0 3149M  4008     0 S  0.7  0.0  6h5
 2705 thomas     20   0 8263M 3989M 50276 S  0.7 50.9  0:0
31235 thomas     20   0 1288M  193M 41516 S  0.7  2.5  4:2
 2852 thomas     20   0  878M  123M 63756 S  0.7  1.6  0:0
 9001 thomas     20   0 1137M 18872  6900 S  0.0  0.2 22:5
32266 thomas     20   0  811M  132M 34880 S  0.0  1.7  0:4
 8751 thomas     20   0 3280M 83640 43324 S  0.0  1.0  1h4
31250 thomas     20   0 1288M  193M 41516 S  0.0  2.5  1:3
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8
```

Figure 5: A possible bug in the software causes the system resources to go completely out of control after closing a vault.

We also observed this behavior in the Windows variant. Although it did not use all of the CPU capacity, the load on our test system still increased permanently to about 30 percent.

Conclusions

Cryptomator significantly increases your data security in the cloud. It relies on intelligent encryption to not only modify the content by cryptographic means, but also abstract the file details and metadata. The software is extremely stable and can also handle large files.

The very simple front end is noteworthy. It does not present any surprises even for inexperienced users, which makes transparent encryption easy to handle even for beginners.

Annoyingly, what appears to be a bug in the software causes the CPU to use up to 100 percent of its capacity when displaying the password prompt on Ubuntu. On openSUSE, the CPU consumption doesn't increase that much, but the RAM is depleted relatively quickly. ■■■

Info

[1] Cryptomator: *https://cryptomator.org*

[2] Scrypt technology: *https://en.wikipedia.org/wiki/Scrypt*

[3] Download Cryptomator: *https://cryptomator.org/downloads/#winDownload*

Cross-platform file compression

Zipline

If you use file compression regularly, zip belongs in your Linux toolbox. *By Bruce Byfield*

File compression in Linux is usually handled by the native commands `tar`, `gzip`, or `bzip2`. However, an additional alternative is `zip` [1], a popular cross-platform command supported by a variety of scripts and utilities. If you are dealing with someone using another operating system, `zip` is often the ideal choice among these compression tools.

Author

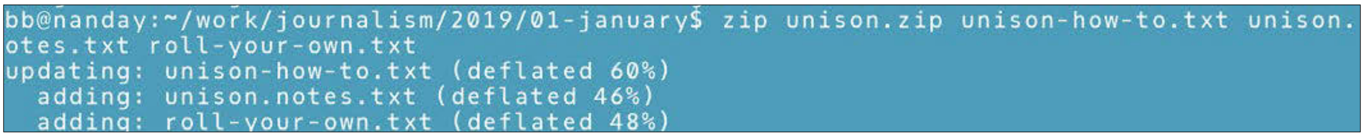
Bruce Byfield is a computer journalist and a freelance writer and editor specializing in free and open source software. In addition to his writing projects, he also teaches live and e-learning courses. In his spare time, Bruce writes about Northwest coast art (*http://brucebyfield.wordpress.com*). He is also co-founder of Prentice Pieces, a blog about writing and fantasy at *https://prenticepieces.com/*.

Admittedly, on Linux, `zip` has fallen out of favor, because for a time it did not support 64-bit computing and could not handle files large than 2MB. Today, though, `zip`, `gzip`, and `bzip2` are broadly similar in functionality and structure. All three have similar options, although not always the same name for every option. All three, too, have a history of providing alternate command names for some functions, such as `unzip` and `ungzip`, that duplicate standard options – presumably to make the commands easier to remember.

Despite these similarities, neither `zip` nor `gzip` recognizes the other's extensions, although both can use files created by the other if the extension is changed. `zip` can use the option `--bzip` to use `bzip` options, which can produce somewhat better compression rates, at least in theory, on binary files.

Zip Basics

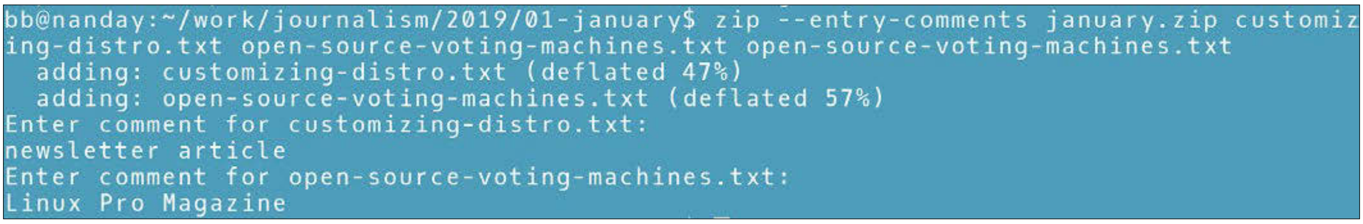
If you have worked with other compression programs, `zip` is easy to start using. The only unusual feature is that a new archive name follows the command and the options and is followed by a space-separated list of the files to archive:



```
bb@nanday:~/work/journalism/2019/01-january$ zip unison.zip unison-how-to.txt unison.
otes.txt roll-your-own.txt
updating: unison-how-to.txt (deflated 60%)
  adding: unison.notes.txt (deflated 46%)
  adding: roll-your-own.txt (deflated 48%)
```

Figure 1: `zip`**'s command structure is simple, but has an unusual order.**

Photo by Alexander Ross on Unsplash



```
bb@nanday:~/work/journalism/2019/01-january$ zip --entry-comments january.zip customiz
ing-distro.txt open-source-voting-machines.txt open-source-voting-machines.txt
  adding: customizing-distro.txt (deflated 47%)
  adding: open-source-voting-machines.txt (deflated 57%)
Enter comment for customizing-distro.txt:
newsletter article
Enter comment for open-source-voting-machines.txt:
Linux Pro Magazine
```

Figure 2: `zip` **includes the ability to add single-line comments about each file.**

```
zip OPTIONS NEW-ARCHIVE.zip FILES
```

Files are not deleted when being added to an archive (Figure 1).

If you choose, the options can include the option `--recurse-patterns` (`-R`), which automatically includes subdirectories. You can also strike a balance between the speed and efficiency of compression with `-NUMBER`, with `0` indicating no compression, `1` the fastest but least compression, and `9` the slowest but greatest compression. The highest compression is somewhere between 2:1 or 3:1 for text, while binary files are usually considerably less, perhaps 3:2 or 4:3.

However, these options are only the beginning. You can use `--exclude` (`-x`) to list files that should not go into the archive or `--include` (`-i`) to specify that only certain files are included. For security, you might also want to add `--password` (`-P`) `STRING` or `--encrypt` (`-e`), although the security is somewhat weak by modern standards (see `zipcloak` below). Still another option is `--entry-comments` (`-c`), which lets you annotate each file in an archive with a single-line comment that can be read using the `zipnote` utility (Figure 2). As the options grow, you are well-advised to add `--test` (`-T`) to ensure that nothing unexpected happens.

`zip`'s options really come into their own once an archive is created. You can use `--delete` (`-d`) or `unzip` to remove files from an existing archive and `--grow` (`-g`) to add files. Alternatively, you can use `--copy-entries` (`-U`) to create a new archive consisting of files in an existing one. If you are creating archives for backups, you can use `--update` (`-u`), `--filesync` (`-FS`), or `--freshen` (`-f`) to keep the backups current. Should an archive become corrupt, you can try to repair it using `--fix` (`-F`). However, many of `zip`'s options require considerably more information. Consequently, the most that I can do in this article is indicate the possibilities.

Zip Utilities

A series of small scripts and utilities have sprouted up around `zip`. These utilities' usefulness is sometimes limited by their inability to handle files larger than 2GB. This limitation extends to larger files made with recent versions of `zip`, including those created from a desktop environment. However, even with these limitations, the utilities can sometimes be useful, especially if you are working with text files. Some are installed alongside `zip`, while one or two have to be installed separately.

zipcmp

Like `diff` with text files, `zipcmp` compares the files in `zip` archives. The command is so simple that it does not include a separate man page or help option. All you need to do is enter the command, followed by the two files to compare. The first file is marked by a minus sign and the second by a plus sign (Figure 3).

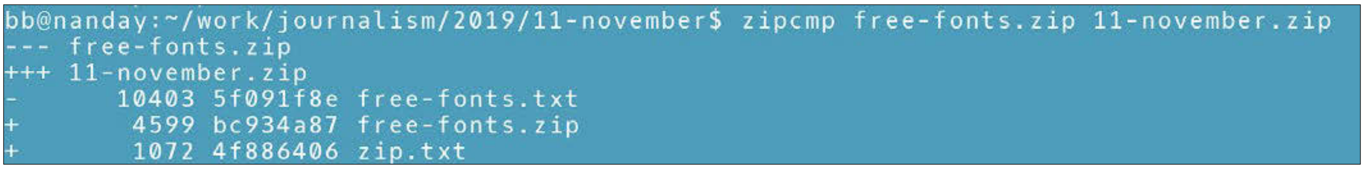
zipcloak

You can password-protect and encrypt `zip` files using `zip --encrypt (-e) option OUTPUT` or `zipcloak OUTPUT (Figure 4)`. If an unencrypted file has been added to the archive, then once you enter the password, the unencrypted file is encrypted. The unencrypted output file is saved to the original directory unless the `-O` command is used to specify another location. With the `-d` option, the entire archive is decrypted, or, if you do not have the password, then copied instead. Because of `zipcloak`'s age, it works only with archives of less than 2GB.

The man page explains that encryption is done through a public domain program that offers a low level of encryption. However, the file that details the program no longer seems to be on the Internet. In the absence of more information, you can still rely on the password for privacy.

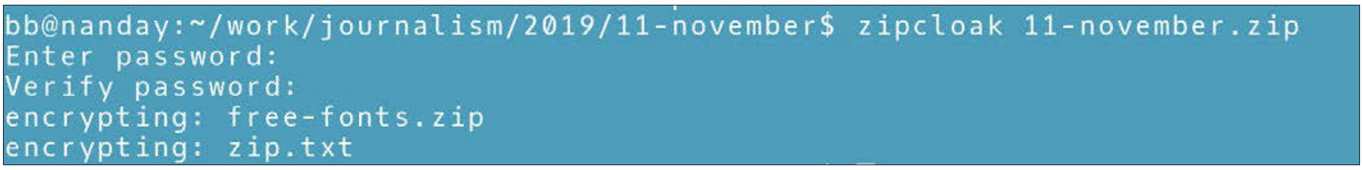
zipgrep

`zipgrep` searches for patterns within a `zip` archive. It can use any pattern supported by `egrep`, such as `*` for any number of characters or `?` for a single charac-



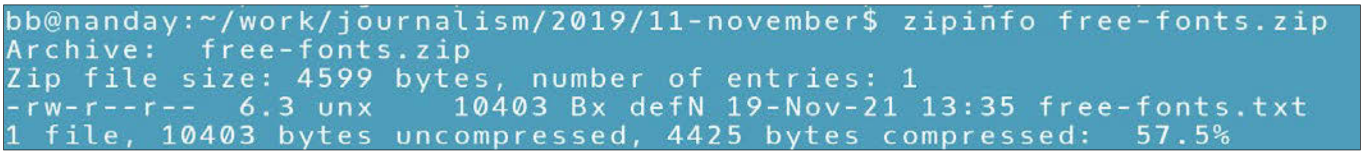
```
bb@nanday:~/work/journalism/2019/11-november$ zipcmp free-fonts.zip 11-november.zip
--- free-fonts.zip
+++ 11-november.zip
-       10403 5f091f8e free-fonts.txt
+        4599 bc934a87 free-fonts.zip
+        1072 4f886406 zip.txt
```

Figure 3: `zipcmp` **compares the files in an archive.**



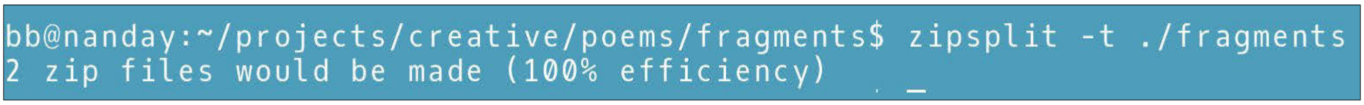
```
bb@nanday:~/work/journalism/2019/11-november$ zipcloak 11-november.zip
Enter password:
Verify password:
encrypting: free-fonts.zip
encrypting: zip.txt
```

Figure 4: **Use** `zipcloak` **to manage encryption on** `zip` **files.**



```
bb@nanday:~/work/journalism/2019/11-november$ zipinfo free-fonts.zip
Archive:  free-fonts.zip
Zip file size: 4599 bytes, number of entries: 1
-rw-r--r--  6.3 unx    10403 Bx defN 19-Nov-21 13:35 free-fonts.txt
1 file, 10403 bytes uncompressed, 4425 bytes compressed:  57.5%
```

Figure 5: `zipinfo` **provides a convenient viewer for the attributes of files in a** `zip` **archive.**



```
bb@nanday:~/projects/creative/poems/fragments$ zipsplit -t ./fragments
2 zip files would be made (100% efficiency)
```

Figure 6: `zipsplit` **divides an archive into smaller ones.**

ter. `zipgrep` cannot be used to search for file names, only the contents of files.

zipinfo

`zipinfo` displays the attributes of all the files in a `zip` archive. This information includes permissions, name, date created, uncompressed and compressed file sizes, and the percentage created (Figure 5).

zipmerge

`zipmerge` adds a source archive into a target archive. Use `zipmerge` carefully, since files in the target will be overwritten by files of the same name in the source. You can reduce the chances of accidental overwriting by using the `-i` option to verify each overwriting. Similarly, `-S` will automatically prevent overwriting files with the same size and name on both the source and target. Also avoid using `-I` to ignore letter case.

Note that `zipmerge` is not installed automatically. When archives merge, both the original files remain, and the only indication of the merge is that the target file is larger.

zipnote

`zipnote` is used to read and edit comments in archives. To read an archive's comment, pipe it to `standard output` with:

```
zipnote FILE-NAME.zip > FILE-NAME.tmp
```

Open `FILE-NAME.tmp` to read and edit. Then add the updated comment with:

```
zipnote -w FILE-NAME.zip < FILE-NAME.tmp
```

`zipnote` is particularly useful for changing file names in an archive that start with `@home` in the comment. However, remember to keep the original extensions.

zipsplit

`zipsplit` divides existing archives into smaller ones, creating an index file to help reassemble them (Figure 6). You might want to divide an archive for two reasons: To reduce any loss of data due to corruption and to store the files across multiple volumes. The default file size is 36KB, but it can be edited using the `-n` option, although the maximum size is 2GB. If you do edit the file size, then you should probably add `-t` to do a dry run and see how many files you will create. Use `zipmerge` to reassemble the original archive. As an alternative, you can use the `--split-size` option.

The output will consist of eight character files, each ending in a number. The first seven characters will be the first seven letters of the original file name.

A Comprehensive Solution

This article barely scrapes the surface of `zip`'s capabilities. `zip` is a command almost as old as personal computing, and its man page consists of over 1,900 lines. Still, even this overview gives some impression of its functionality. Even though it is not a traditional part of the Unix toolkit, if you use file compression regularly, `zip` is worth a closer look. And should you be sending files to some other operating system, your colleagues will thank you for using it. ■■■

Info

[1] zip: *https://linux.die.net/man/1/zip*

The sys admin's daily grind: Delivery notification

Hey, Mr. Postman

Our columnist Charly Kühnast doesn't want to be out when a package delivery service is standing at his front door with some goodies. When the courier rings, he can have the Telegram API send a notification to his mobile phone. *By Charly Kühnast*

I mainly find notifications on my mobile phone annoying and have disabled most of them. But things are different when something requires my attention at home. The classic example is the doorbell: I want to know when a delivery person is standing at my front door even if I'm flooding my noise-killing headset with electric guitar music. I already solved this problem back in 2016 with a Raspberry Pi and described the setup in this column [1].

With my earlier solution, notifications reached me as Twitter direct messages, mainly because I didn't want to install a notification app just for this purpose. Twitter is already installed on my phone and provides a useful programming interface. However, Twitter is increasingly setting up hurdles for access to the API. For example, you now need to describe in a small, free text essay the purpose of your proposed use of the API. This may be understandable from the supplier's point of view, but I would prefer things to be a little simpler.

This explains why I now use the Telegram API [2]. Among other things, I can create bots that can be used to do many funny things. On top of this, I can get the bot to send me messages. You can create a bot by starting a chat with *BotFather* in the Telegram app (Figure 1).

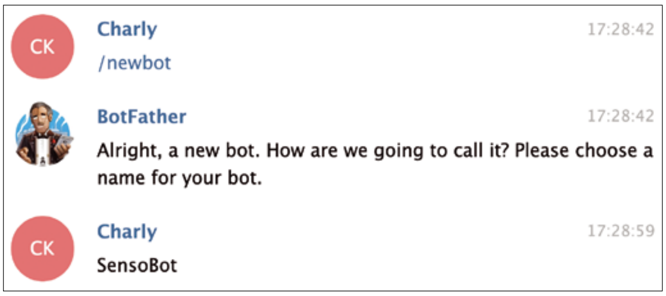
Start with the `/newbot` command. BotFather asks you for a name and a username for your bot. In return, you are sent a numeric ID and an access token – that is, a string that you have to pass to the API to identify yourself. It could look like this, for example:

```
919445171:↩
  AAFt4RTVEcI919fAI_5YGHXOA8QyQnY_fBA
```

The first few digits are your bot's numeric ID, followed by a colon and the access token. Now you have to start a chat with your bot in the app or the web interface [3]. This chat also has a numeric ID, which you need to discover. The `curl` command from Listing 1 does the trick.

Note that you need to prefix your bot's numeric ID with the `bot` keyword. For the most part, the long answer is unimportant for the purposes of this article, but it does contain a string reading `"id":930313820` – this is the chat ID. Now you have all the information you need.

If you now want to automatically send a message to the bot chat, using a Bash script for example, the `curl` call from Listing 2 is all it takes. The phone vibrates and a message appears on the display (Figure 2). Now, I'll never miss my deliveries again. ■■■

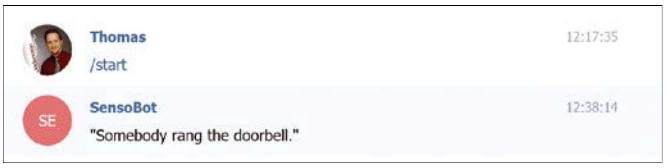


Charly 17:28:42
/newbot

BotFather 17:28:42
Alright, a new bot. How are we going to call it? Please choose a name for your bot.

Charly 17:28:59
SensoBot

Figure 1: Create a new bot by opening a conversation with the *BotFather*.



Thomas 12:17:35
/start

SensoBot 12:38:14
"Somebody rang the doorbell."

Figure 2: When the delivery person rings the bell, the bot is triggered and my phone vibrates.

Info

[1] "The sys admin's daily grind: Charly's doorbell Pi" by Charly Kühnast, *Linux Magazine* , issue 186, May 2016, p. 62, *http://www.linux-magazine.com/Issues/2016/186/Charly-s-Column-Doorbell-Pi/(language)/eng-US*

[2] Telegram API: *https://core.telegram.org/api*

[3] Telegram web interface: *https://web.telegram.org*

Author

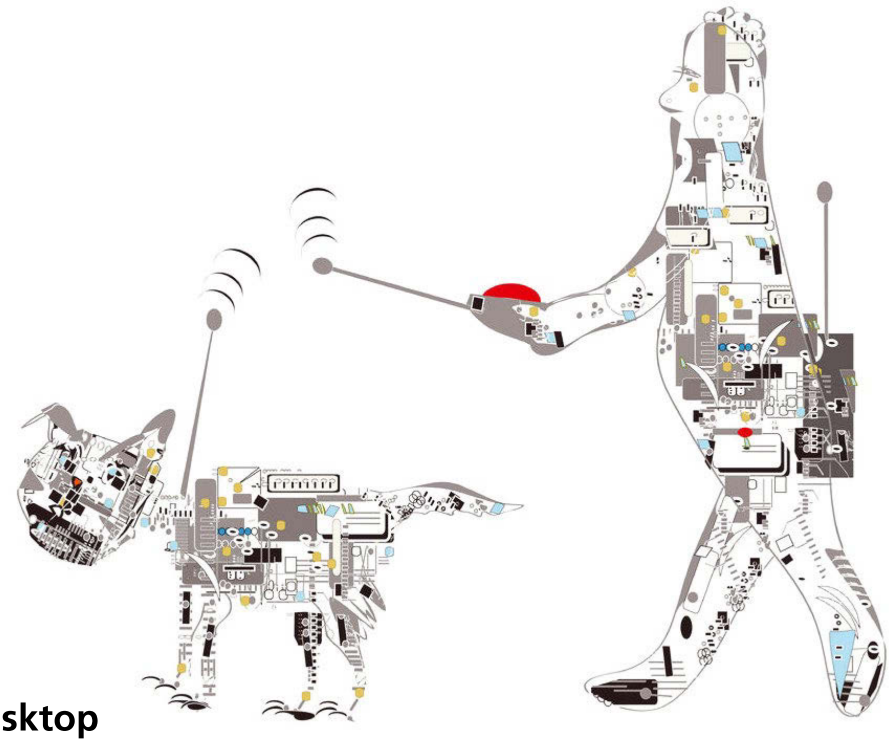
Charly Kühnast manages Unix systems in a data center in the Lower Rhine region of Germany. His responsibilities include ensuring the security and availability of firewalls and the DMZ.

Listing 1: Start Telegram Chat

```
$ curl -X POST https://api.telegram.org/bot919445171:AAFt4RTVEcI919fAI_5YGHXOA8QyQnY_fBA/getUpdates
```

Listing 2: Sending a Message to the Bot Chat

```
$ curl -X POST 'https://api.telegram.org/bot919445171:AAFt4RTVEcI919fAI_5YGHXOA8QyQnY_fBA/sendMessage?chat_
              id=930313820&text="Hello world!"'
```



Control your Android smartphone from your desktop

Remote Control

With guiscrcpy graphical user interface and the scrcpy command-line tool, you can control your smartphone from the comfort of your Linux desktop and even record what you're doing onscreen. *By Christoph Langner*

Smartphones and the Linux desktop complement each other quite well thanks to cloud services. If you're not afraid of Google services or providers like Dropbox, WhatsApp, and the like, you never again have to connect your mobile phone to your computer to synchronize data or copy photos and videos. Even messengers like WhatsApp or Google's Messages now offer a web interface so that your mobile phone can stay in your pocket.

However, there are also situations in which you might want to be able to use your PC to operate your mobile phone. Ideally, you would want to transfer the content of the mobile phone display to the PC desktop. This is exactly what the duo of the `scrcpy` command-line program and the guiscrcpy graphical user interface (GUI) offers. These open source programs available for Linux, macOS X, and Windows even let you record what is happening on the mobile phone display.

Screen Copy

`scrcpy` comes from the Genymobile [1] treasure trove. Among other things, Genymobile specializes in virtualizing Android, which means that developers can test their applications on a variety of smartphones or tablets without having to own the devices themselves. As a command-line tool, Genymotion provides the `scrcpy` [2] program's source code on GitHub for download under the very liberal Apache license.

Developed independently of `scrcpy`, the guiscrcpy GUI [3] adds some convenience. The program integrates all the `scrcpy` functions and lets users open the connection with a single click. Freely placeable toolbars with buttons on the desktop help to transmit typical actions, such as the *Home*, *Back*, and *Menu* buttons, as well as, for example, rotating the smartphone's display.

Installation

Both guiscrcpy and `scrcpy` are still so new that they are not available in the major distributions' package sources. Only Arch Linux allows you to install the two programs from the Arch User Repository (AUR). Listing 1 shows you how to import guiscrcpy from the AUR; alternatively you can use another AUR helper or graphical package manager like Manjaro's Pamac. When installing from the AUR, the system automatically fetches `scrcpy` and the other dependencies to your machine.

For other distributions (like Ubuntu 19.04, which we use in our lab), the developer offers precompiled binary packages together with an installation script. To install, download the current version (1.10.1 when this issue went to press) in the form of a `tar.gz` archive, unpack the file as shown in Listing 2, and execute the installation script. It saves the program as `guiscrcpy` and also creates an entry in the Application menu. (To uninstall guiscrcpy, see the "Deinstallation" box.)

Listing 1: Importing from the AUR

```
$ yay -S guiscrcpy wmctrl xdotool
```

Deinstallation

Guiscrcpy's developer currently provides an installation script, but there is no option to delete the program from the system. Use the commands from Listing 3 to remove the application and the configuration file from the home directory, if needed.

Lead Image © victor kuznetsov, 123RF.com

18 YEARS OF LINUX! Issues 1-214
Searchable Linux Archive: 19,000 Pages of Practical Know-How
► All the Tricks
► All the Hacks
► All the Apps
ISSUE 215 OCT 2018
LINUX MAGAZINE
DVD ROM
THE COMPLETE LINUX MAGAZINE ARCHIVE

Listing 2: Installing guiscrcpy

```
$ tar xzf guiscrcpy-1.10.1-05092019-linux.tar.gz
$ sudo ./guiscrcpy-systemless-linux-installer.sh
$ guiscrcpy
```

Listing 3: Removing guiscrcpy

```
$ sudo rm /usr/local/bin/guiscrcpy
$ sudo rm -rf /usr/share/guiscrcpy
$ sudo rm /usr/share/applications/guiscrcpy*.desktop
$ rm ~/local/guiscrcpy.cfg
```

Listing 4: Installing scrcpy and adb

```
$ sudo snap install scrcpy
$ sudo apt install android-tools
```

Listing 5: Checking for Attached Devices

```
$ adb devices
List of devices attached
LGH87094110190 device
```

Guiscrcpy functions as the interface for the `scrcpy` command-line tool. `scrcpy` is also missing from popular distributions' package sources, but the developers do offer the program as a Snap package [4] – many distributions now support this format out the box. On Ubuntu 19.04, you can import the tool with Snapcraft using a single command. In the same step, you also install the Android Debug Bridge (`adb`) via the Android Software Development Kit provided by Google (Listing 4).

Developer Mode

For guiscrcpy to access your smartphone, developer mode and USB debugging must be enabled on the mobile device. Google has hidden the settings you need for this. To unlock them, open *Settings* on your smartphone, then in *General* | *About Phone* | *Software Information* tap the *Build Number* seven times. Depending on the smartphone manufacturer, the menu path may vary.

You now should find a new *Developer Options* menu item in the settings, where you can enable both the developer options en bloc and *USB Debugging* as an additional option (Figure 1). Later, the smartphone will ask you if you want to allow USB debugging when you connect via guiscrcpy for the first time. You can use a fingerprint to identify the connected computer later on and save the connection using the *Always allow from this computer* option.

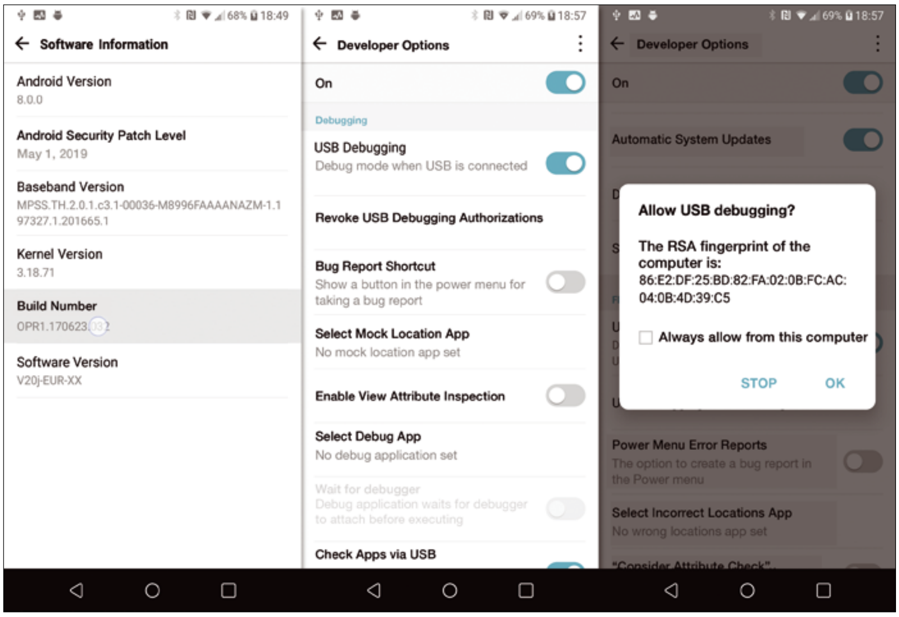
Remote Control

With guiscrcpy, you can open the connection to the smartphone directly with the *START SCRCPY* button (Figure 2). The program should automatically detect the Android smartphone or tablet connected via a USB cable. The *DEVICE LGH87094110190 IS CONNECTED* field is where the application reports the mobile phone it detected. If problems occur detecting the device, run the `adb devices` command in a terminal window to directly check whether the system has really found the smartphone (Listing 5).

Guiscrcpy transmits the content of the phone display at a bit rate of up to 16000KBps (8000KBps in the default setting, which is totally fine). As options, you can enable a full-screen mode (*Fullscreen*), *Always on Top*, and *Keep display off (only screen mirroring)*. The latter option turns off the device's screen during transmission, but the touch screen input still works.

If you are making a video that demonstrates an Android app (or if you as a developer want to explain an app function to your users), you can record the events on the display with the *Record screen* option. The recording is stored as an MP4 file in the current user's home directory. The *Show touches* option is also available for recordings: It visualizes finger taps on the display with a circle.

To capture actions typical of Android (such as pressing the *Home*, *Menu*, or *Back* buttons), guiscrcpy displays horizontal and vertical toolbars (Figure 3). You can use these buttons to change the display orientation, adjust the audio playback volume, or exchange texts between the smartphone and Linux desktop via the clipboard. You can position the toolbars freely on the



Software Information

Android Version
8.0.0

Android Security Patch Level
May 1, 2019

Baseband Version
MPSS.TH.2.0.1.c3.1-00036-M8996FAAAANAZM-1.1 97327.1.201665.1

Kernel Version
3.18.71

Build Number
OPR1.170623.032

Software Version
V20j-EUR-XX

Developer Options

On

Debugging

USB Debugging
Debug mode when USB is connected

Revoke USB Debugging Authorizations

Bug Report Shortcut
Show a button in the power menu for taking a bug report

Select Mock Location App
No mock location app set

Enable View Attribute Inspection

Select Debug App
No debug application set

Wait for debugger
Debug application waits for debugger to attach before executing

Check Apps via USB

Developer Options

On

Automatic System Updates

Allow USB debugging?

The RSA fingerprint of the computer is:
86:E2:DF:25:BD:82:FA:02:0B:FC:AC:04:0B:4D:39:C5

☐ Always allow from this computer

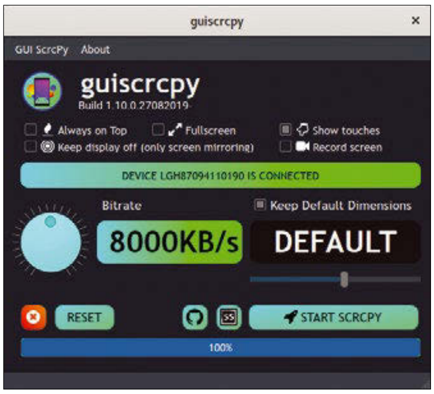
STOP OK

Power Menu Error Reports
The option to create a bug report in the Power menu

Select Incorrect Locations App
No wrong locations app set

"Consider Attribute Check"

Figure 1: To allow guiscrcpy to access the screen of an Android smartphone connected via USB, you need to enable developer mode and USB debugging.



guiscrcpy

GUI ScrcPy About

guiscrcpy
Build 1.10.0.27082019

☐ Always on Top ☐ Fullscreen ☑ Show touches
☐ Keep display off (only screen mirroring) ☐ Record screen

DEVICE LGH87094110190 IS CONNECTED

Bitrate 8000KB/s

☑ Keep Default Dimensions DEFAULT

RESET START SCRCPY

100%

Figure 2: All the functions of the `scrcpy` command-line tool can be controlled using the guiscrcpy GUI. Click *START SCRCPY* to initiate the connection.

desktop by holding down the right mouse button.

Conclusions

Installing guiscrcpy, `scrcpy`, and Android tools is still a little more complicated than necessary, but the program comes with major benefits for Android developers in particular. With a few clicks, you can create videos of your smartphone screen, with which you can demonstrate your program's functions. The achievable data rate is so high that even videos played on a mobile phone can be played back on the desktop PC with practically no jitter (Figure 4).

Even non-programmers will find uses for guiscrcpy. Not every program offers a web interface like WhatsApp or Messages. Since guiscrcpy forwards mouse and keyboard input from the PC to the mobile phone, you can write a text message for an Android app on your desktop and forward it to your smartphone. ■■■

Info

[1] Genymobile: *https://www.genymobile.com*

[2] scrcpy: *https://github.com/genymobile/scrcpy*

[3] guiscrcpy: *https://srevinsaju.github.io/guiscrcpy/*

[4] Snap for scrcpy: *https://snapcraft.io/scrcpy*

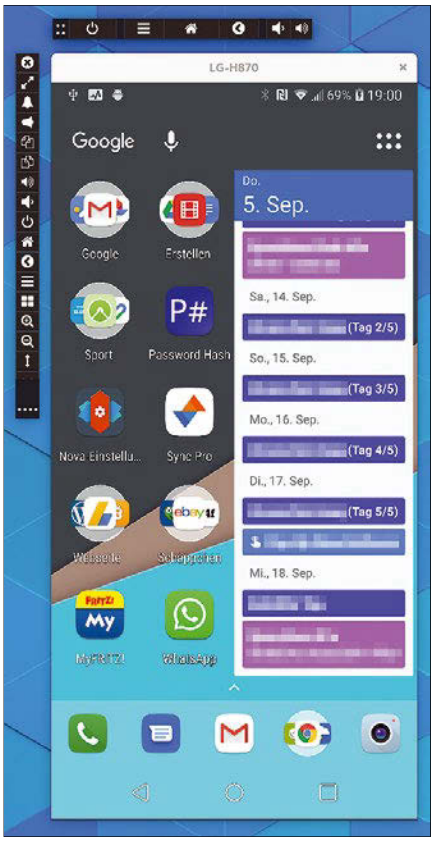
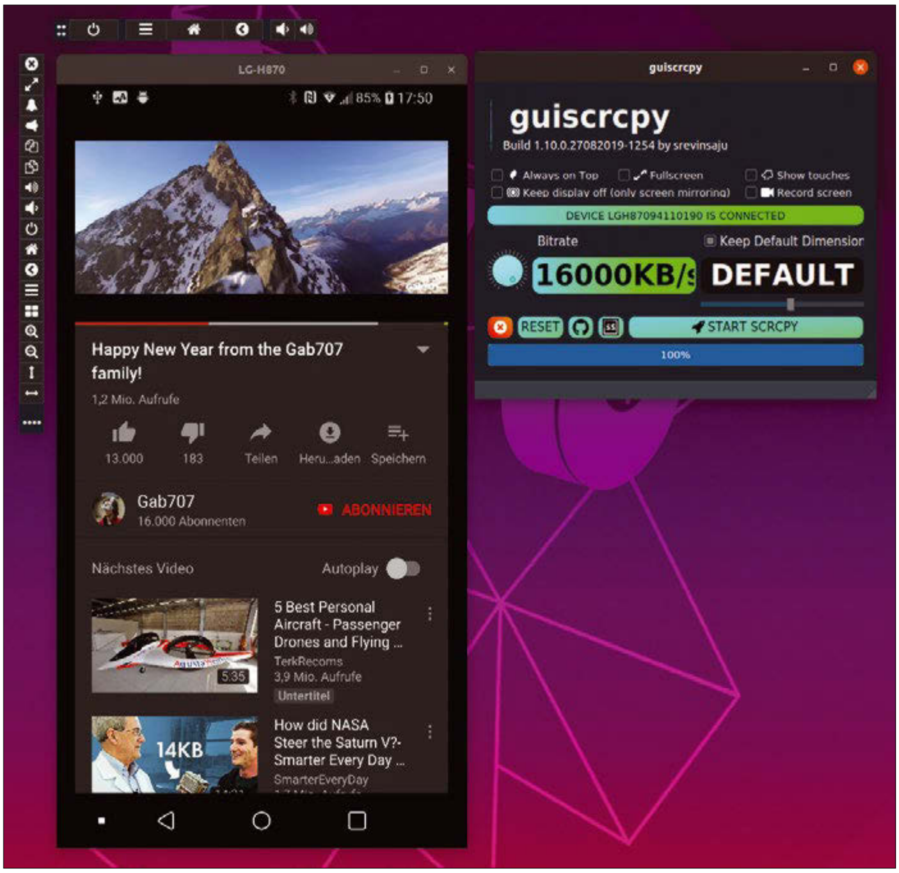


Figure 3: An Android smartphone screen on the Linux desktop: Guiscrcpy transfers mouse and keyboard input directly to the mobile phone. The toolbars can be used to trigger additional actions, such as rotating the display.



guiscrcpy
Build 1.10.0.27082019-1254 by srevinsaju

□ Always on Top □ Fullscreen □ Show touches
□ Keep display off (only screen mirroring) □ Record screen

DEVICE LGH87094110190 IS CONNECTED

Bitrate: 16000KB/s
□ Keep Default Dimension: DEFAULT

RESET
START SCRCPY
100%

Figure 4: Even videos played on the smartphone appear practically jitter-free on the Linux desktop.

Displaying Docker containers and their history with Go

Monitoring Station

With a monitoring system implemented in Go, Mike Schilli displays the Docker containers that have been launched and closed on his system. *By Mike Schilli*

Even in the age of newfangled buzzwords like "Kubernetes," veteran system administrators still appreciate command-line tools like good old `top`, which displays running processes in real time. Since I also want to keep up with the young bloods (even at my age, which can already be described as advanced), I recently created a small, terminal-based monitor that shows Docker containers coming and going on a system.

The standard Docker client, `docker`, is written in Go and communicates via a web interface with the Docker daemon to query the status of running containers, start new ones, or terminate existing ones. In addition to a Python interface, Docker also offers a Go SDK. Since Go also has excellent libraries for displaying in the terminal, it was my choice for implementing the `dockertop` monitor in this Programming Snapshot column.

The idea is simple: The program asks the Docker daemon at regular intervals for all the containers running on the system, displays their names in a list, and refreshes them every second – just like `top`. As an additional treat, the monitor in the right part of the split screen also displays a rolling history of the containers. Each time it detects a new one, the program writes `New: <Name>` into it. If a container has been lost since the last call – for example, because it has died in the meantime – the log entry reads: `Gone: <Name>` (Figure 1).

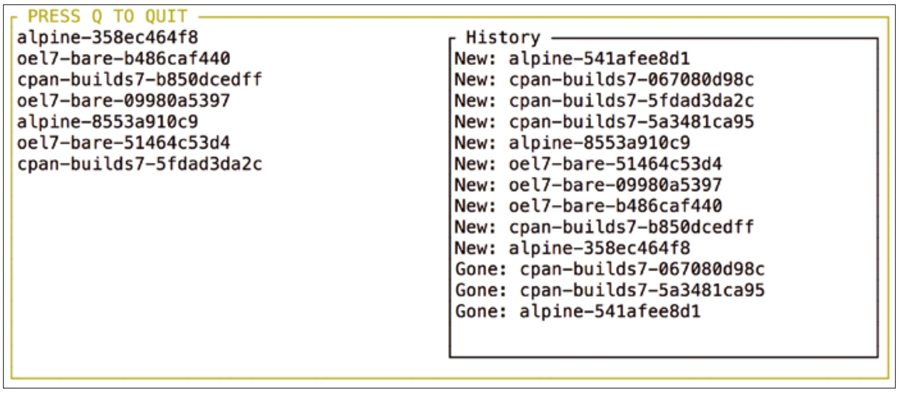
This gives the sys admin an impression, even on a busy system with many containers, of how the individual instances are doing. Depending on how fast the log changes, you can guesstimate whether there is a problem causing the started container to fail immediately or if everything is within bounds.

Courtesy of Google

This Programming Snapshot column has introduced terminal user interfaces (UIs) [1] several times in previous editions, including `termui` [2] and `promptui` [3]. This time it's a Google framework based on `termui`: Termdash, which is especially suitable for the dashboards of the data world.

Listing 1 [4] implements the graphical components of the terminal UI in Figure 1 and adds a whole litany of Go libraries from GitHub. Since the widget named `container` in line 7 would collide with the Docker containers used later, the code fetches the component under the name `tco`. The somewhat verbose handling of individual errors in Go is accelerated by the `panicOnError()` function starting at line 16. On a production system, the code would probably handle errors explicitly and in a dedicated way, instead of immediately aborting the program if something goes wrong. But in our example, this saves us a long listing.

The `context` construct created in line 28 is a kind of remote control that subroutines pass on to each other in Go. If the main program calls the returned `cancel()` function, this signals the end to the context, and all subrou-



```
┌ PRESS Q TO QUIT ──────────────────────────────────────────────────┐
│alpine-358ec464f8                    ┌ History ──────────────────┐│
│oel7-bare-b486caf440                 │New: alpine-541afee8d1     ││
│cpan-builds7-b850dcedff              │New: cpan-builds7-067080d98c││
│oel7-bare-09980a5397                 │New: cpan-builds7-5fdad3da2c││
│alpine-8553a910c9                    │New: cpan-builds7-5a3481ca95││
│oel7-bare-51464c53d4                 │New: alpine-8553a910c9     ││
│cpan-builds7-5fdad3da2c              │New: oel7-bare-51464c53d4  ││
│                                     │New: oel7-bare-09980a5397  ││
│                                     │New: oel7-bare-b486caf440  ││
│                                     │New: cpan-builds7-b850dcedff││
│                                     │New: alpine-358ec464f8     ││
│                                     │Gone: cpan-builds7-067080d98c││
│                                     │Gone: cpan-builds7-5a3481ca95││
│                                     │Gone: alpine-541afee8d1    ││
│                                     └───────────────────────────┘│
└───────────────────────────────────────────────────────────────────┘
```

Figure 1: The left column shows the active Docker containers, while the right column has the history of lost and found containers.

Author

Mike Schilli works as a software engineer in the San Francisco Bay area, California. Each month in his column, which has been running since 1997, he researches practical applications of various programming languages. If you email him at *mschilli@perlmeister.com* he will gladly answer any questions.

Lead Image © Erik Reis, 123RF.com

tines get the message and can initiate cleanup actions.

The application's main window contains two text windows side by side, as seen in Figure 1. The `top` widget displays the list of active containers similar to the `top` Unix utility, while the rolling log window (`rolled`) to the right provides the historical view of containers coming and going. To arrange them side by side, the code employs the helpers `Left()` and `Right()` with a call to `SplitVertical()` in the terminal. When the user presses *Q*, you want Go to clear the UI and abort the program. This is why line 56 defines in `quit` a callback of the keyboard watchdog that triggers when the user presses the correspond-

Listing 1: dockertop.go

```
01 package main
02
03 import (
04   "context"
05   "fmt"
06   "github.com/mum4k/termdash"
07   tco "github.com/mum4k/termdash/container"
08   "github.com/mum4k/termdash/linestyle"
09   "github.com/mum4k/termdash/terminal/termbox"
10   "github.com/mum4k/termdash/terminal/terminalapi"
11   "github.com/mum4k/termdash/widgets/text"
12   "strings"
13   "time"
14 )
15
16 func panicOnError(err error) {
17   if err != nil {
18     panic(err)
19   }
20 }
21
22 func main() {
23   t, err := termbox.New()
24   panicOnError(err)
25   defer t.Close()
26
27   ctx, cancel :=
28    context.WithCancel(context.Background())
29
30   top, err := text.New()
31   panicOnError(err)
32
33   rolled, err := text.New(
34     text.RollContent(), text.WrapAtWords())
35   panicOnError(err)
36
37   go updater(top, rolled)
38
39   c, err := tco.New(
40     t,
41     tco.Border(linestyle.Light),
42     tco.BorderTitle(" PRESS Q TO QUIT "),
43     tco.SplitVertical(
44       tco.Left(
45         tco.PlaceWidget(top),
46       ),
47       tco.Right(
48         tco.Border(linestyle.Light),
49         tco.BorderTitle(" History "),
50         tco.PlaceWidget(rolled),
51       ),
52     ),
53   )
54   panicOnError(err)
55
56   quit := func(k *terminalapi.Keyboard) {
57     if k.Key == 'q' || k.Key == 'Q' {
58       cancel()
59     }
60   }
61
62   err = termdash.Run(ctx, t, c,
63     termdash.KeyboardSubscriber(quit))
64   panicOnError(err)
65 }
66
67 func updater(top *text.Text,
68              rolled *text.Text) {
69   items_saved := []string{}
70   for {
71     err, items, _ := dockerList()
72     panicOnError(err)
73
74     add, remove :=
75       diff(items_saved, items)
76
77     for _, item := range add {
78       err := rolled.Write(
79         fmt.Sprintf("New: %s\n", item))
80       panicOnError(err)
81     }
82     for _, item := range remove {
83       err := rolled.Write(
84         fmt.Sprintf("Gone: %s\n", item))
85       panicOnError(err)
86     }
87
88     content := strings.Join(items, "\n")
89     if len(content) == 0 {
90       content = " " // can't be empty
91     }
92     err = top.Write(content,
93       text.WriteReplace())
94     panicOnError(err)
95
96     items_saved = items
97     time.Sleep(time.Second)
98   }
99 }
```

ing key. Once in action, the callback in line 58 calls the `cancel()` function of the previously created context, which in turn triggers lower-level cleanup functions.

For the UI to be able to react to changes in the context, the object is passed to the UI main loop starting with `Run()` in line 62, along with a list of all widgets to be used. When it's time to close shop, the UI's internal main event loop detects this via the passed in context and neatly winds down the UI. Without a controlled exit, the program would leave the terminal in graphics mode, in which case the user would no longer be able to enter shell commands or get a proper prompt. Closing the terminal window and opening a new one is usually the only way out of a mess like this.

Groundhog Day

The Go routine `updater()` called asynchronously from line 37 defines the time loop that refreshes the UI with the latest data from the Docker daemon every second. Starting at line 67, it fetches the list of containers via `dockerList()`, which I'll get to in a bit in Listing 2. The left subwindow with the `top` view refreshes itself with the call to `top.Write()` in line 92 in Listing 1 with a long `content` string containing the individual container names with 10 characters of their ID, separated by line breaks.

Containers that the monitor sees for the first time are reported by the `diff()` function called in line 75 of Listing 1. You'll see its inner workings later in Listing 4, but for now it just returns two array slices, `add` and `remove`, which are generated from the difference between the last container listing (`items_saved`) and the current one (`items`). All these steps are embedded in an endless `for` loop, at the end of which, in line 97, the call to `time.Sleep()` pauses for one second before it enters the next round. The loop and the `sleep` command run in a Go routine (i.e., asynchronously), and thus the UI remains fully responsive.

That was it for the UI, whose implementation neatly fits into 99 lines. So how does the Go program get access to the active containers' names on the system? The Docker API's individual components and their functions are described in great detail on the project's website, which has a link to automatically generated documentation from comments in the Go source code [5].

However, with its open source Moby project, Docker has cooked up a strange brew here and does not follow the versioning common in the Go community. Consequently, the otherwise successful `go mod init`, which is used to prepare Listing 2 for compilation by fetching the source code from GitHub during the build phase, does not work. Instead, the user has to install the library (Listing 3, line 1) and repeat the process with all libraries pulled in by `import` statements in the listings. Only then can you build the `dockertop` binary (line 2). If you used the modern module method, it would fail, because the Docker API delivers an ancient version that does not support some functions used in the listings.

Hello Daemon, Client Speaking

As a simple Docker client, which fetches the list of all containers from the daemon, `docker ps` called from the shell would also be useful; its standard output would dump out the names. Instead, I'm using the Docker Client API – because I can, and because it can later be extended at will – but it takes a little more effort.

Line 19 in Listing 2 creates a new client object and passes the parameter `WithAPIVersionNegotiation` to it. This is enormously important: Without it, the client on a somewhat outdated Ubuntu system complains that the server is rejecting it, because the client version number is supposedly too high. But passing the version negotiation parameter fixes the problem, and both start talking to each other. `ContainerList()` returns a list of active container objects, sorted by start date. The Docker image for each container can be found in the `.Image` attribute and will be displayed in the UI alongside the container ID.

In order for the client to be able to distinguish between several Ubuntu containers running in parallel, line 34 uses `container.ID[:10]` to add the first 10 charac-

Listing 2: dockerlist.go

```
01 package main
02
03 import (
04   "context"
05   "fmt"
06   "github.com/docker/docker/api/types"
07   "github.com/docker/docker/client"
08 )
09
10 func dockerList() (error, []string,
11     map[string]types.Container) {
12   items := []string{}
13   containerMap :=
14     make(map[string]types.Container)
15
16   opt :=
17     client.WithAPIVersionNegotiation()
18   cli, err :=
19     client.NewClientWithOpts(opt)
20   if err != nil {
21     return err, nil, nil
22   }
23   defer cli.Close()
24
25   containers, err := cli.ContainerList(
26       context.Background(),
27       types.ContainerListOptions{})
28   if err != nil {
29     return err, nil, nil
30   }
31
32   for _, container := range containers {
33     name := fmt.Sprintf("%s-%s",
34       container.Image, container.ID[:10])
35     items = append(items, name)
36     containerMap[name] = container
37   }
38
39   return nil, items, containerMap
40 }
```

Listing 3: Building dockertop

```
$ go get -u github.com/docker/docker/client
$ go build dockertop.go dockerlist.go dockerdiff.go
```

ters of the container's unique ID. The names of all containers found in this way are appended to a slice of strings in line 35, so that the original order in which the server reported them is retained.

Additional information on each container ends up in the `containerMap` attribute under `items`. This allows other program parts to access the correctly sorted list, as well as more details if required. `dockerList()` returns both data structures to the caller.

What Changed?

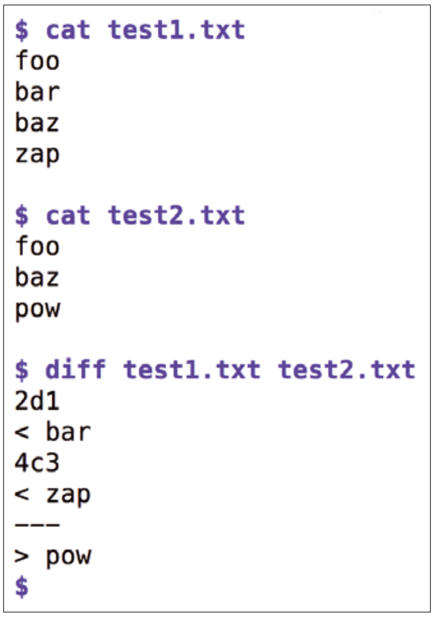
Finally, the program maintains in its right subwindow a history log of all containers that appear and later go away. For this, Listing 4 compares two string array slices in the `diff()` function and determines which new entries have been added in the second array and which entries are in the first array but have not made it into the second.

Unix experts know the procedure from the `diff` tool, which also displays the difference between two files in terms of lines that have either been added or omitted. Figure 2 illustrates that `diff` correctly finds out that the `bar` and `zap` entries have been removed from the `test1.txt` file and that the `pow` entry has been added to `test2.txt`.

Smart Algorithm

How does the `diff` algorithm work? The basis is the longest common subsequence (LCS) procedure [6], which determines the longest common partial sequence of two arrays. A naive approach could simply delete all entries from the first array and add all items of the second array to arrive at the result. But that's not the point: It's about getting from A to B with as few steps as possible.

The LCS method provides a series of identical positions in both strings. In this way, it determines that the first entry `foo` occurs at the same position in both strings. The `baz` entry, on the other hand, has been shifted: In the first string, it is at the third position; in the second, it is at the second position (index *(2,1)*). The LCS algorithm there-



```
$ cat test1.txt
foo
bar
baz
zap

$ cat test2.txt
foo
baz
pow

$ diff test1.txt test2.txt
2d1
< bar
4c3
< zap
---
> pow
$
```

Figure 2: The `diff` algorithm is based on the LCS method.

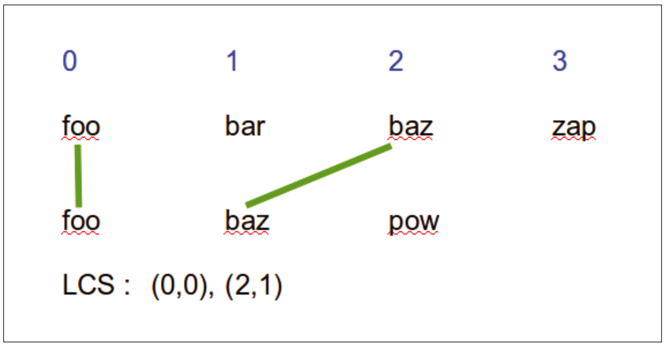


Figure 3: LCS pairs for the arrays from Figure 2.

fore outputs the pairs *(0,0),(2,1)* when comparing the files `test1.txt` and `test2.txt`, as shown in Figure 3.

Listing 4 retrieves the LCS algorithm from the `golcs` project on GitHub. From the delivered pairs, the code calculates container names to `add` and to `remove`. In order to move from the last list to the current one, it traverses the old and new arrays (`left` and `right`) step by step using `for` loops, while moving from pair to pair.

One difficulty here is the strict type system in Go: The LCS algorithm on GitHub was implemented with a generic type (`interface{}`), similar to `void` in C, because it is supposed to be able to analyze data of any type. To be able to process arrays of strings, however, the programmer needs to convert them into arrays of `interface{}` types first, which is laborious and time-consuming. Otherwise the compiler refuses to call the library function. The reason given by Go followers is that strings have a different memory layout than `interface{}` types. This mess will hopefully be resolved by the next Go version.

Conclusions and Outlook

Granted, the procedure for determining the Docker history is not 100 percent accurate. Between two queries, the Docker daemon could have created a container that disappeared on the second call. A ghost container of this kind could only be detected by a subscription mechanism that would receive a message from the Docker daemon on each event. Maybe I'll add it in the next iteration of this project.

The procedure is still good enough to observe appearing and vanishing containers and will help to detect any irregularities. Since it is a DIY program, there are no limits to the developer's creativity: Click on a displayed container and it shuts down? Sort by alphabetical order rather than by start date? Make missing containers red and new ones green? As always, the solution is just a few keystrokes away. ■■■

Info

[1] "A Go terminal UI for displaying network adapters in real time" by Mike Schilli, *Linux Magazine*, issue 218, January 2019, pp. 42-45

[2] "Calculating weekdays and dates with Go" by Mike Schilli, *Linux Magazine* issue 227, October 2019, pp. 44-47

[3] "Go program stores directory paths" by Mike Schilli, *Linux Magazine*, issue 228, November 2019, pp. 42-45

[4] Listings for this article: *ftp://ftp.linux-magazine.com/pub/listings/linux-magazine.com/231/*

[5] Docker Client API: *https://godoc.org/github.com/docker/docker/client#Client*

[6] LCS: *https://en.wikipedia.org/wiki/Longest_common_subsequence_problem*

Listing 4: dockerdiff.go

```
package main
import "github.com/yudai/golcs"

func diff(old []string,
          new []string) (add []string,
            remove []string) {

  left := make([]interface{}, len(old))
  for i, v := range old {
    left[i] = v
  }
  right := make([]interface{}, len(new))
  for i, v := range new {
    right[i] = v
  }

  l := lcs.New(left, right)

  leftidx := 0
  rightidx := 0

  for _, pair := range l.IndexPairs() {
    for leftidx < len(left) &&
        leftidx <= pair.Left {
      if leftidx < pair.Left {
        remove =
          append(remove, old[leftidx])
      }
      leftidx++
    }
    for rightidx < len(right) &&
        rightidx <= pair.Right {
      if rightidx < pair.Right {
        add = append(add, new[rightidx])
      }
      rightidx++
    }
  }

  for leftidx < len(left) {
    remove = append(remove, old[leftidx])
    leftidx++
  }
  for rightidx < len(right) {
    add = append(add, new[rightidx])
    rightidx++
  }

  return add, remove
}
```

Data synchronization over the Internet with Unison

IN SYNC

Unison and SSH help you sync your directories across multiple computers. *By Werner Hein*

Not everyone is lucky enough to get through life with a single IT device for everything. Many Linux users have a distributed IT infrastructure, with a PC at home, another PC at work, and a laptop for the road.

In such a constellation, you always want to be able to access the current version of all files, regardless of which device you are currently using.

A typical solution to this dilemma is to take a working copy of all the relevant files with you, for example, on a USB stick. But if you don't pay meticulous attention, you could end up with several diverging versions of a file, none of which contains all the changes.

Only centralized data storage on a network drive in the cloud can help. However, the access speed and thus also your efficiency can suffer enormously. In addition, the client must be able to reach the server on the Internet at all times, which is often a problem when commuting or traveling.

For a really practicable solution to the problem, you need three components to work together: centralized data storage that is accessible on the Internet, local working copies of this pool on all devices, and a simple – preferably fully automated – method for synchronizing via a server and clients. Unison [1] is a popular open source tool that can handle the synchronization duties. This article shows how you can combine Unison with the OpenSSH secure shell [2] to connect your local systems to the cloud and synchronize your files (Figure 1).

Setting Up SSH

SSH provides the basis for authenticating the client to the server and for transporting the data over the Internet with protection from unauthorized access.

See the project website for more on setting up OpenSSH [2]. On the server, the SSH service must listen on a defined port (typically port 22). If the server is behind a firewall or a router, you will need to forward the port.

Each client must have a secret key and its public counterpart. A typical scenario is to use an RSA key pair consisting of two files: `~/.ssh/id_rsa` and `~/.ssh/id_rsa.pub`. If these keys are missing, or if you have forgotten the secret key's passphrase, you can generate a new key pair with `ssh-keygen`. To create a new passphrase, simply call `ssh-keygen -p`. You'll need to communicate each client's public key to the server. For example, if the server is located in the `mycloud.com` domain, you could enter:

```
$ $ ssh-copy-id ↩
client_name@server_name.mycloud.com
```

Photo by Jason Hafso on Unsplash

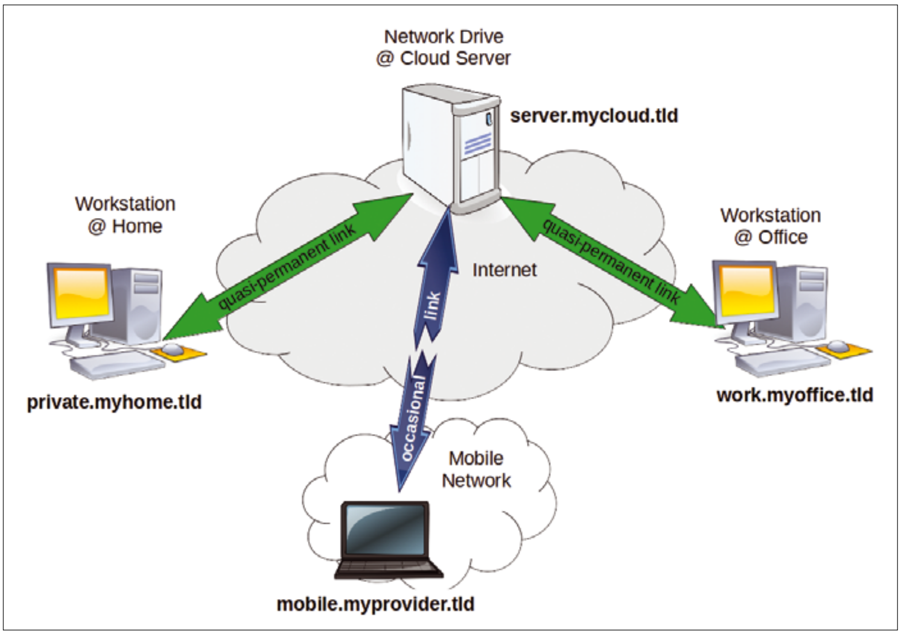


Figure 1: The user files on three computers, two with a quasi-permanent and one with an occasional network connection, are synchronized over the Internet against a central server.

The key ends up in the `~/.ssh/authorized_keys` file on the server. From now on, the way to connect from the client to the server is to enter the local secret key's passphrase; you won't have to rely on the remote server password.

To enter this passphrase only once per session on the client, use `ssh-agents`; this command launches automatically at login time on many current desktop systems. If you pass in the secret key unlocked by passphrase via `ssh-add`, then no further typing of the passphrase is required for each subsequent SSH connection during this session. You will therefore also want to run `ssh-add` at login by adding it to an autostart script.

Desktops typically use the graphical `ssh-askpass` dialog to query the passphrase. Some desktops, such as KDE Plasma, use their own implementation (for Plasma: `ksshaskpass`), which allows them to store the passphrase in the integrated password safe for further sessions after entering it. This step eliminates the need to re-enter the passphrase in the future.

The commands for installing OpenSSH on the server and the corresponding client on desktops and notebooks appear in Listing 1.

Listing 1: Installing OpenSSH

```
[... Server ...]
$ sudo apt-get install openssh-server
[... Client (KDE Plasma) ...]
$ sudo apt-get install openssh-client ksshaskpass
[... Client (andere Desktop-Umgebungen) ...]
$ sudo apt-get install openssh-client ssh-askpass
```

Setting Up Unison

For Unison to work properly, all the systems involved in replication – all the clients and the central server – must use the same software version. Both Unison's User Manual and Reference Guide refer explicitly to the fact that the data format of the archives can change between releases. Therefore it is recommended to generate Unison directly from the source code in order to be able to mix different distributions and their generations more easily in the working environment. This article uses the currently recommended stable Unison v2.48.4 [3].

To build Unison, you need GNU Make, the OCaml compiler, and some libraries. To install them on Debian and its offshoots including all necessary dependencies, use the command from line 1 of Listing 2. Then change to the home directory, download the Unison source code, unpack it, and change to the newly created source directory (lines 2 to 5).

Essentially, Unison consists of three executable files, which you build with the commands from lines 6 to 10 and install to `/usr/local/bin` (lines 11 to 17). The command-line-based Unison performs the actual sync and can be used both in the background and interactively at the command line. Finally, the `unison-fsmonitor` filesystem monitor is used in case of a permanent

Listing 2: Building Unison

```
01 $ sudo apt-get install make ocaml libgtk2.0-dev liblablgtk2-ocaml-dev
02 $ cd
03 $ wget --content-disposition https://github.com/bcpierce00/unison/archive/2.48.4.tar.gz
04 $ tar xvzf unison-2.48.4.tar.gz
05 $ cd ~/unison-2.48.4
06 $ make UISTYLE=text
07 $ sudo mv src/unison /usr/local/bin/unison-2.48.4
08 $ sudo mv src/unison-fsmonitor /usr/local/bin/
09 $ make UISTYLE=gtk2
10 $ sudo mv src/unison /usr/local/bin/unison-2.48.4-gtk
11 $ cd /usr/local/bin
12 $ sudo chown root:root unison-*
13 $ sudo chmod 0755 unison-*
14 $ sudo ln -s unison-2.48.4 unison-2.48
15 $ sudo ln -s unison-2.48 unison
16 $ sudo ln -s unison-2.48.4-gtk unison-2.48-gtk
17 $ sudo ln -s unison-2.48-gtk unison-gtk
```

Listing 3: common.prf

```
01 # Unison common profile to sync any client's
02 # home directory with a server on the internet
03
04 root = /home/<Client>
05 root = ssh://<Client>@<server>.<mycloud>.<tld>:<SSH-Port>//home/<Client>
06
07 logfile = /home/<Client>/.unison/unison.log
08
09 ignore = Path Downloads
10 ignore = Path Maildir
11 ignore = Name .*
12
13 ignorenot = Path .unison
14 ignore = Path .unison/*
15 ignorenot = Path .unison/*.prf
16
17 ignorenot = Path .gnupg
18 ignore =  Path .gnupg/*
19 ignorenot = Path .gnupg/{gpg.conf,pubring.gpg,secring.gpg}
20
21 ignorenot = Path .ssh
22 ignore = Path .ssh/known_hosts*
```

connection between client and server. When changes are made on one page, the client informs the other side and initiates a sync immediately.

The interactive alternative with a graphical user interface (GUI) is `unison-gtk`. It is especially recommended for mobile devices with only a temporary connection to the server and in case of inconsistencies between client and server. Inconsistencies inevitably arise as soon as files on both sides change for some reason without a timely sync being able to take place. Unison can no longer resolve this automatically. In such situations, the graphical client is clearer, making it easier to select the appropriate file version.

Configuring the Synchronization

The way Unison works is controlled by profile files, which can be nested if required (more about this later). These profiles follow semantics, the fundamentals of which I will explain with some examples. When you edit the files, be sure to consult the Unison manual, especially the chapters on profiles [4] and path specification [5].

As a userspace application, Unison stores its configuration in the `~/.unison` directory; it is the individual user's responsibility to design this to suit their own needs. Consequently, the Unison profile `common.prf` (Listing 3) can only serve as an example; you will need to adapt it to meet your specific needs.

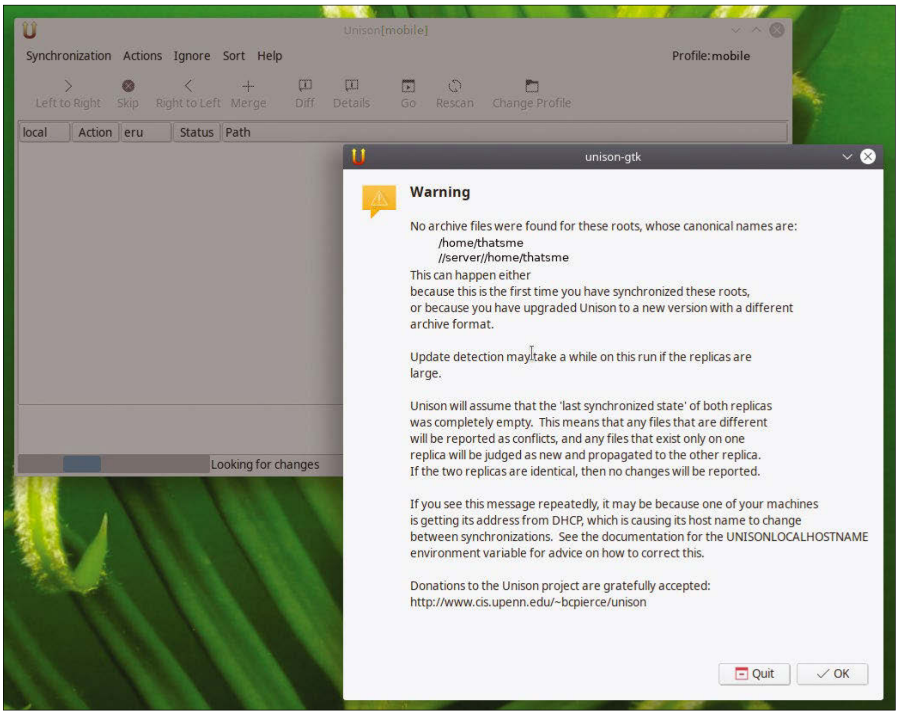
Line 4 in Listing 3 specifies the local directory that Unison will synchronize against the server and its directory (line 5). Line 7 defines the logfile that logs all individual adjustments.

In line 9, things start to get more interesting: Lines 4, 5, and 7 tell Unison to compare the entire content of the specified directories against each other. Lines 9 and 10 explicitly exclude two paths in the client root directory (`/home/<Client>`), `Downloads/` and `Maildir/` in this case, from the synchronization.

In addition, line 11 excludes all files and directories that have the specified pattern in their name. In this case, this means all the hidden files and directories (dotfiles) in the entire path from the root directory, for example, including the file `/home/<Client>/<Folder>/.<File>`. This also includes the `/home/<Client>/.unison/` directory and its contents.

However, Unison needs to keep its profiles – only the ones from the `.unison/` directory – in sync. This explains why line 13 first includes `.unison/` again, but line 14 excludes its content (logs and archives). Line 15 then ensures that all files ending in `.prf` (i.e., the Unison profiles) are still synchronized.

Lines 17 to 19 do something similar for the hidden `.gnupg/` directory. However, line 19 here explicitly includes only the three listed files in the synchronization.



Warning

No archive files were found for these roots, whose canonical names are:
    /home/thatsme
    //server//home/thatsme
This can happen either
because this is the first time you have synchronized these roots,
or because you have upgraded Unison to a new version with a different
archive format.

Update detection may take a while on this run if the replicas are
large.

Unison will assume that the 'last synchronized state' of both replicas
was completely empty. This means that any files that are different
will be reported as conflicts, and any files that exist only on one
replica will be judged as new and propagated to the other replica.
If the two replicas are identical, then no changes will be reported.

If you see this message repeatedly, it may be because one of your machines
is getting its address from DHCP, which is causing its host name to change
between synchronizations. See the documentation for the UNISONLOCALHOSTNAME
environment variable for advice on how to correct this.

Donations to the Unison project are gratefully accepted:
http://www.cis.upenn.edu/~bcpierce/unison

Figure 2: During the first sync, a warning appears. You can only press *Ok* here in an ideal case (i.e., if the client has no files).

Listing 4: mobile.prf

```
01 # Unison profile for mobile
02 # clients with occasional
03 # network connection
04
05 # explicitly exclude use
06 # of file system monitor
07 watch = false
08
09 # load definitions for
10 # actual reconciliation
11 include common
```

First Sync

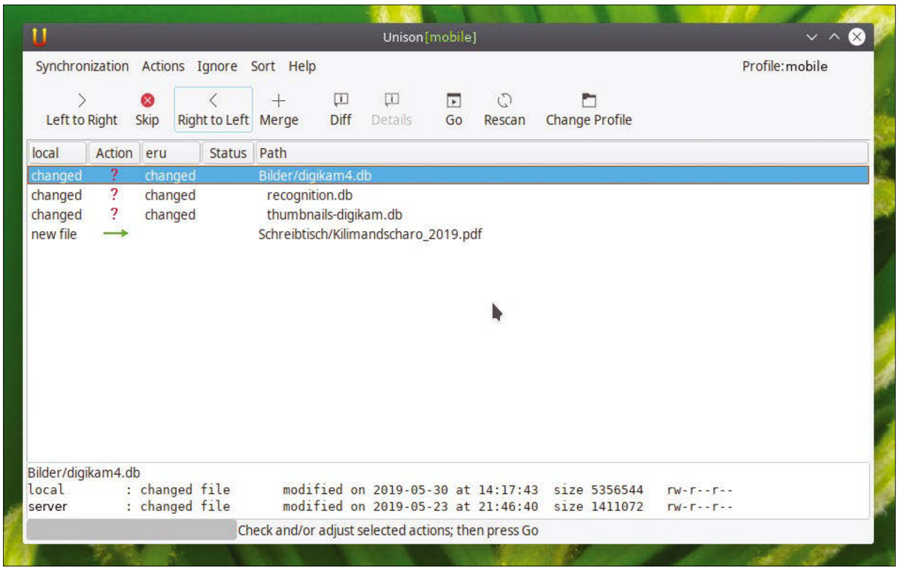
For an initial sync, I will be assuming the following initial scenario. The file tree to be synchronized is completely filled on the server; on the client, it is completely empty, except for the Unison profiles in `~/.unison/` and the SSH certificate in `~/.ssh/`. In this ideal state, the server thus exclusively acts as the source, and the client exclusively acts as the sync target.

If another state exists, you need to prepare for a longer session, because then many conflicts can occur on the first attempt. All of them need to be handled with the greatest care – otherwise you could easily shoot yourself in the foot. Under no circumstances must the client contain the remains of any Unison sync archives in the directory `~/.unison/`.

You trigger this initial sync in the same way as for mobile devices in the following section. You will then see the warning shown in Figure 2 in response. In it, Unison announces that this is obviously an initial sync, which could take a little longer.

Mobile Operation

The mode for mobile operation is mainly – but, as explained later, not exclusively – intended for devices like the `mobile.myprovider.tld` laptop in Figure 1. For this simple operating mode, a quite straightforward profile (Listing 4) is all you need; you can save this as `mobile.prf`. It ensures that the `unison-fsmonitor` filesystem monitor does not launch under any circumstances (line 7) and that only the basic sync takes place in line with the specifications from Listing 3 (line 11 in Listing 4).

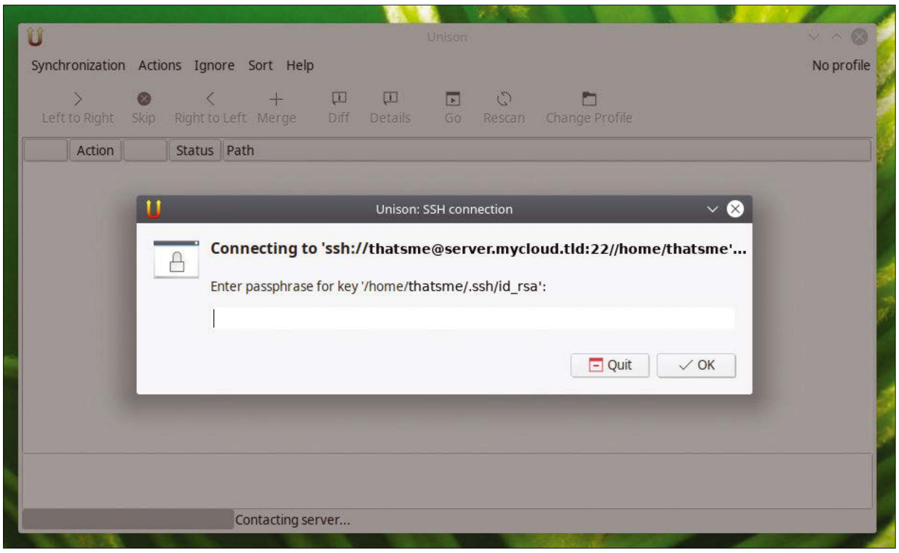


local	Action	eru	Status	Path
changed	?	changed		Bilder/digikam4.db
changed	?	changed		recognition.db
changed	?	changed		thumbnails-digikam.db
new file	→			Schreibtisch/Kilimandscharo_2019.pdf

```
Bilder/digikam4.db
local         : changed file      modified on 2019-05-30 at 14:17:43  size 5356544   rw-r--r--
server        : changed file      modified on 2019-05-23 at 21:46:40  size 1411072   rw-r--r--
```

Check and/or adjust selected actions; then press Go

Figure 3: A typical representation of the state between two replicas in the Unison GUI.



Connecting to 'ssh://thatsme@server.mycloud.tld:22//home/thatsme'...

Enter passphrase for key '/home/thatsme/.ssh/id_rsa':

Contacting server...

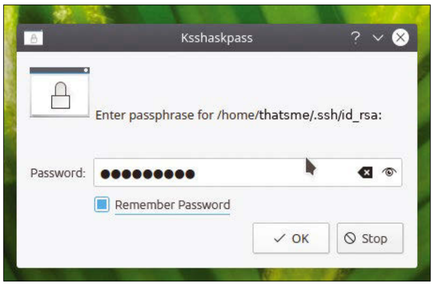
Figure 4: Without further preparation, Unison directly prompts you for the passphrase for the SSH connection.

To start the synchronization, it makes sense to launch the graphical application. At the command line, type `unison-gtk mobile`. Since running the program from the prompt is not entirely intuitive, you will want to create an appropriate shortcut on the desktop, in the taskbar, or in the application menu.

After starting Unison, wait until the application shows you the differences between the two replicas (Figure 3). In the simplest case, if the sync suggested by the green arrows is what you expected, then press *Go* and wait for the end of the synchronization run. Even in the case of conflicts marked with red question marks, you can start the sync with a clear conscience: Unison then only processes the updates with the green arrows and leaves the conflicts untouched.

Listing 5: unison-mobile.sh

```
#!/bin/sh

# Load user's SSH Certificate into SSH agent

# Hint: Passphrase will be requested

# -- either to be entered manually

# -- or will be retrieved from desktop's wallet

ssh-add < /dev/null > /dev/null 2>&1
```



Enter passphrase for /home/thatsme/.ssh/id_rsa:

☑ Remember Password

Figure 5: Transferring the passphrase to the desktop wallet (here on KDE Plasma) saves typing for future syncs.

Authentication

Whenever it is called, Unison prompts you for the passphrase of the certificate for the SSH connection (Figure 4). As long as you do not close the `unison-gtk` user interface after you have entered your data, you can perform as many syncs as you like without having to re-enter it. This is a pretty ugly hack, though.

It makes far more sense to integrate the `unison-mobile.sh` shell script in Listing 5 into the desktop environment so that it is called automatically at login time. Then the passphrase request is only made once in order to release the SSH certificate for the server for the duration of the session. It is then available for any number of syncs – and for regular SSH sessions.

If you not only enter the passphrase, but also check the *Save Password* box as shown in Figure 5, the desktop stores the passphrase in its wallet, and you will not need to query it again for future logins.

Handling Conflicts

After synchronization, the Unison user interface displays the results in a clear-cut overview (Figure 6). Successfully synchronized files are marked with a green check mark; a summary lists the remaining conflicts. If you now press *Rescan* and compare the replicas again, all the successfully matched files will disappear and only the conflicts will remain.

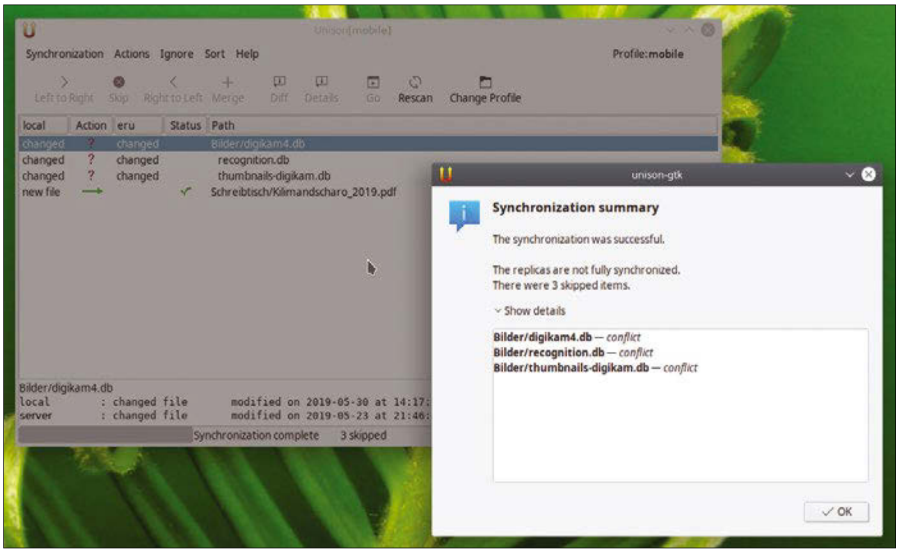
Select the conflicts one by one and prepare them for a new sync by selecting *Skip*, *Left to Right*, or *Right to Left*. Unison shows the approach chosen for conflict resolution with blue arrows (Figure 7). Another click on *Go* then triggers the sync.

Stationary Operation

The `stationary.prf` profile (Listing 6) is intended for computers with a permanent network connection, such as the workstations `private.myhome.tld` and `work.myoffice.tld` (see Figure 1).

Activating batch mode and other modes in lines 9 to 11 means that Unison runs completely in the background, no longer asks you any questions, and synchronizes everything that it can without a conflict occurring. Line 12 additionally switches on the `unison-fsmonitor` automatic filesystem monitor in the background, which ensures a quick comparison as soon as something changes in one of the two replicas.

Due to the fact that the connection is established during login and is only closed again at logout time, there is per-



```
local    Action  eru      Status  Path
changed  ?       changed          Bilder/digikam4.db
changed  ?       changed          recognition.db
changed  ?       changed          thumbnails-digikam.db
new file →                ✓       Schreibtisch/Kilimandscharo_2019.pdf

Bilder/digikam4.db
local     : changed file      modified on 2019-05-30 at 14:17:
server    : changed file      modified on 2019-05-23 at 21:46:
Synchronization complete    3 skipped
```

Synchronization summary

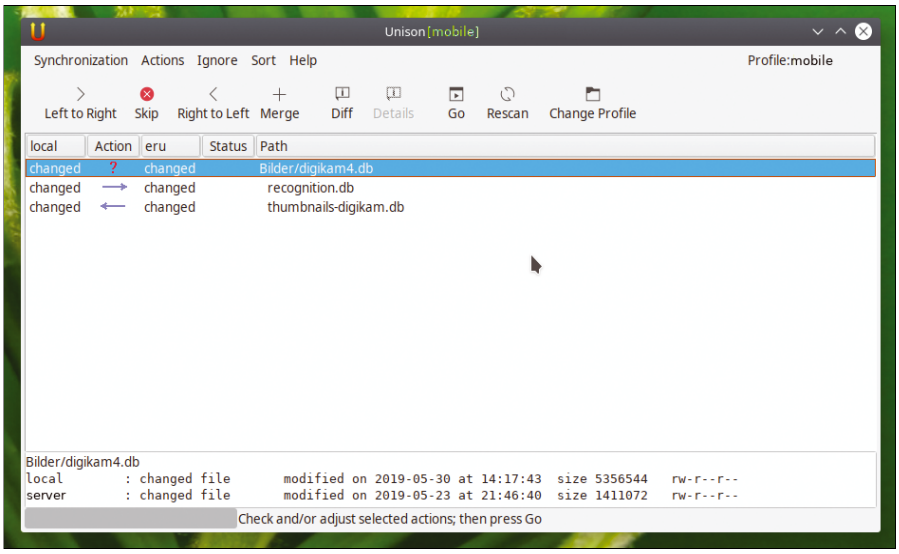
The synchronization was successful.

The replicas are not fully synchronized.
There were 3 skipped items.

⌄ Show details

Bilder/digikam4.db — *conflict*
Bilder/recognition.db — *conflict*
Bilder/thumbnails-digikam.db — *conflict*

Figure 6: Unison showing the results of a comparison and pointing out conflicts that still exist.



```
local    Action  eru      Status  Path
changed  ?       changed          Bilder/digikam4.db
changed  →       changed          recognition.db
changed  ←       changed          thumbnails-digikam.db

Bilder/digikam4.db
local     : changed file      modified on 2019-05-30 at 14:17:43   size 5356544   rw-r--r--
server    : changed file      modified on 2019-05-23 at 21:46:40   size 1411072   rw-r--r--
Check and/or adjust selected actions; then press Go
```

Figure 7: Conflicts during synchronization are resolved manually by selecting a suitable direction for each synchronization.

Listing 6: stationary.prf

```
01 # Unison profile for stationary
02 # clients with (almost) permanent
03 # network connection
04
05 # don't ask, just constantly
06 # reconcile based on file
07 # watcher by propagating
08 # non-conflicting changes
09 batch = true
10 silent = true
11 auto = true
12 repeat = watch
13
14 # load definitions for
15 # actual reconciliation
16 include common
```

Listing 7: unison-stationary.sh

```
01 #!/bin/sh
02 # Load user's SSH Certificate into SSH agent
03 # and start unison w/ cmdline user interface
04 # Hint: unison runs in background and starts
05 # unison-fsmonitor on client and server side
06 ssh-add < /dev/null > /dev/null 2>&1
07 unison -ui text stationary > /dev/null 2>&1
```

manent communication between client and server. This is handled by the `unison-stationary.sh` shell script (Listing 7), which you need to add to the list of autostart scripts. On KDE Plasma, for example, put it in the `~/.config/autostart-scripts/` directory.

Line 6 in Listing 7 activates the SSH certificate – as above for mobile operation – and loads it for the session. Line 7 starts Unison in text mode and as a background process with the profile from Listing 6. From now on and until you log out, every change in the filesystem on one of the two sides triggers a real-time comparison in the background and is logged in the logfile. Conflicts seemingly disappear, but can be detected at any time by calling `unison-gtk mobile` in the GUI and handled accordingly.

Conclusions

Unison v2.48.4 is a very reliable, powerful, and efficient tool for bidirectional synchronization of filesystems between computers. In combination with SSH, this comparison can also be carried out securely over the Internet with regard to authentication and data transport. Smaller synchronization runs optimize the solution for mobile or stationary operation.

Unison's GUI clearly shows the differences between the two replicas and supports the user in resolving conflicts. With the configuration shown here, databases can be kept up to date across several computers, which allows the user to freely and quickly switch between workstations. ■■■

Author

Werner Hein has been working with Linux in his spare time since 1994 and professionally with the development of modems in mobile phones since 1997. He uses the solution shown here to synchronize his notebook and two PCs at home and at work via a central server in his home LAN.

Info

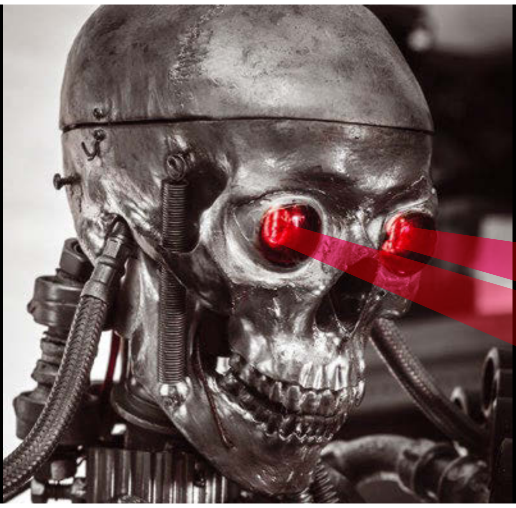
[1] Unison: *https://www.cis.upenn.edu/~bcpierce/unison/*

[2] OpenSSH: *https://www.openssh.com*

[3] Unison 2.48.4 (source code): *https://github.com/bcpierce00/unison/archive/2.48.4.tar.gz*

[4] Profiles: *https://www.cis.upenn.edu/~bcpierce/unison/download/releases/stable/unison-manual.html#profile*

[5] Path specification: *https://www.cis.upenn.edu/~bcpierce/unison/download/releases/stable/unison-manual.html#pathspec*



MakerSpace

Video surveillance with the Raspberry Pi

Eye in the Pi

Set up a professional video surveillance system armed with motionEyeOS and a Raspberry Pi.

By Erik Bärwaldt

Theft, vandalism, and property damage are commonplace occurrences today. Police investigations into such cases often come up empty, and the culprits escape undetected and unpunished. As a result, sales of video surveillance systems consisting of several cameras and a video recorder are booming. Here, I show you how, armed with just a Raspberry Pi and motionEyeOS, you can catch the culprits in the act.

Thanks to inexpensive IP and USB cameras, you can find numerous software packages for the home that can be used to upcycle a computer into a full-fledged video surveillance system. However, these solutions all have a serious disadvantage: They have to run constantly and require a correspondingly large amount of energy.

Professional surveillance cameras are bulky and usually attached by a cable to a control unit; just cutting the data cable will disable the camera. Video surveillance with the Raspberry Pi and the motionEyeOS [1] free operating system, on the other hand, is simple, less conspicuous, and less expensive.

Author

Erik Bärwaldt is a self-employed IT-admin and technical author living in Scarborough (United Kingdom). He writes for several IT-magazines.

Installation

MotionEyeOS is based on BuildRoot and comes as a small source tarball or as images from the developer's website [2] for various Raspberry Pis and other small board computers. After downloading and unpacking, install the system on a microSD card. For example, on Linux, enter:

```
dd if=<Image> of=/dev/<Device> bs=1M
```

Replace the device name with the name of your microSD card, which you can discover by typing the `lsblk` command in a terminal window. Now you can boot the Raspberry Pi from the freshly written microSD card.

Before the first startup of motionEyeOS, connect the cameras intended for video surveillance to the Raspberry Pi through a USB port or connect them to your home network. USB cameras are easier to configure over software, but they require a data cable, making wireless cameras a far better choice for surveillance purposes.

The first deployment requires a wired connection to the router: Without such a connection, the boot process aborts after the automatic network setup fails, and the operating system restarts in an infinite loop.

Once you have fulfilled all the requirements, all the services you need

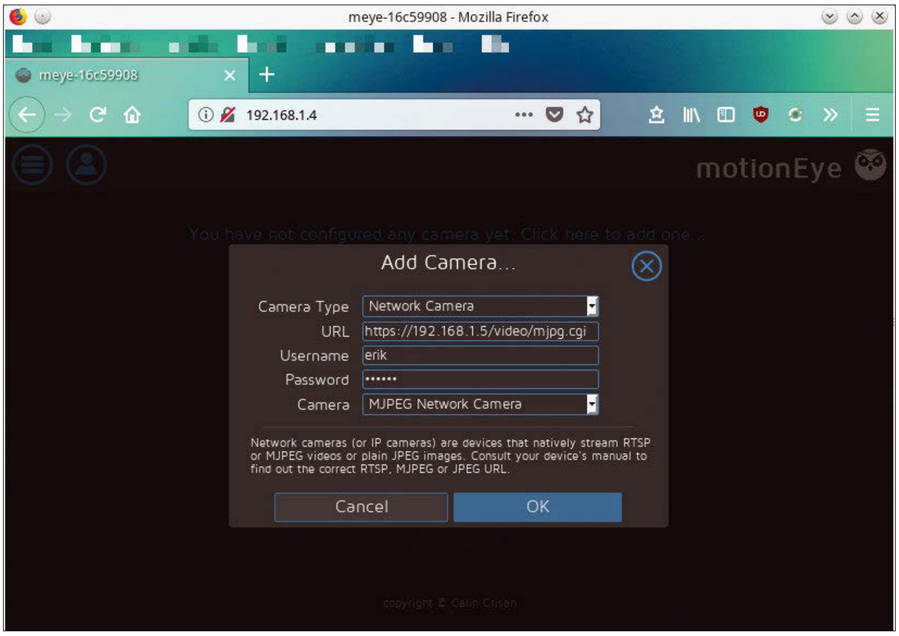
Lead Image © svedoliver, 123RF.com

will start up automatically when the system is booted, and you will see a login prompt. The first time the system boots, it also creates a new data partition automatically on the memory card and formats it. This can take a long time with large memory cards; however, you should use as much memory as possible, because the recordings from all connected cameras end up on this partition.

After starting the system, you can access the IP address of the Raspberry Pi in your web browser from any workstation on your home network. The browser then opens the login screen of the motionEyeOS front end, which is written in Python and acts as the graphical interface for the Motion daemon.

The username is *admin* with no password. The software then prompts you to configure the cameras in an almost empty browser window. Now enter the configuration data for the individual cameras in a small dialog.

The software uses the Video4Linux API to address cameras connected to a USB port. You can connect network cameras to the motionEyeOS system by way of Motion JPEG (MJPEG) or the Real Time Streaming Protocol (RTSP) (Figure 1). To secure the connection, you also need to specify the authentication credentials used to access a wireless or wired camera or your local network.


meye-16c59908 - Mozilla Firefox
meye-16c59908
192.168.1.4
motionEye
You have not configured any camera yet. Click here to add one...

Add Camera...

Camera Type: Network Camera
URL: https://192.168.1.5/video/mjpg.cgi
Username: erik
Password: ••••••
Camera: MJPEG Network Camera

Network cameras (or IP cameras) are devices that natively stream RTSP or MJPEG videos or plain JPEG images. Consult your device's manual to find out the correct RTSP, MJPEG or JPEG URL.

Cancel
OK
copyright © Calin Crisan

Figure 1: In addition to webcams connected by USB, the motionEye system supports IP cameras integrated into the network.

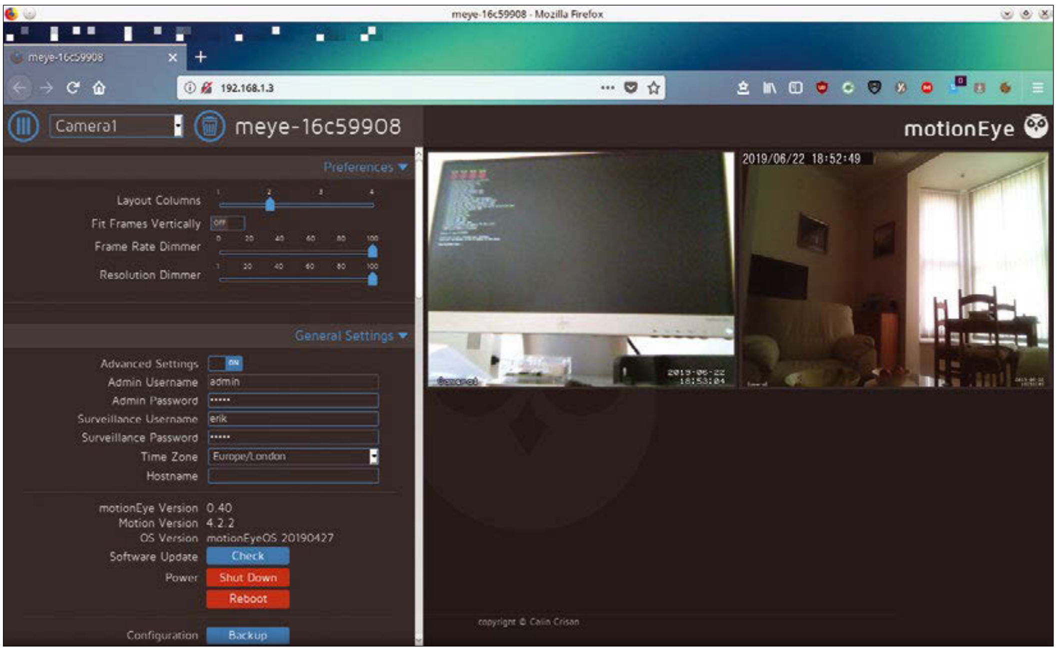
Problematic

When setting up cameras that reside on the local network, motionEyeOS exhibits a minor error, which you can avoid by manually editing a configuration file. With careful configuration, wired and wireless cameras request authentication data during access, rather than being openly accessible in the network.

If you have secured the camera with a self-signed certificate, motionEyeOS simply denies access. In this case, log on to the Raspberry Pi as the administrator over SSH and edit the `/data/` `etc/` file in the `motioneye.conf` directory. Next, add a line reading `validate_certs false` at the end of the file. After a reboot, the software no longer checks for certificates; the camera in question can then talk to the system, and the web browser displays the current camera image (Figure 2).

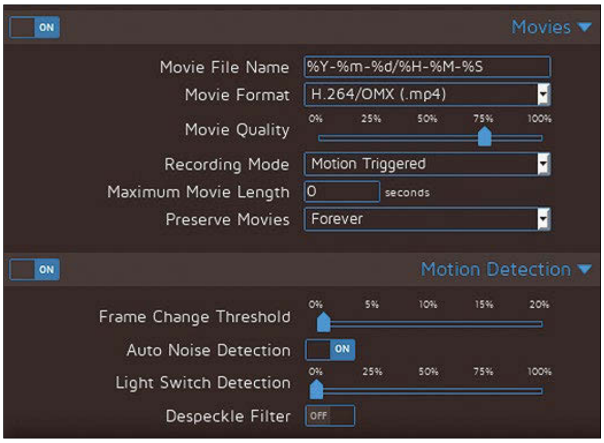
Settings

If you click on the menu icon at top left in the application window, the software opens a configuration menu on the left



Layout Columns: 2
Fit Frames Vertically: OFF
Frame Rate Dimmer: 100
Resolution Dimmer: 100
Advanced Settings: ON
Admin Username: admin
Admin Password: •••••
Surveillance Username: erik
Surveillance Password: •••••
Time Zone: Europe/London
Hostname:
motionEye Version: 0.40
Motion Version: 4.2.2
OS Version: motionEyeOS 20190427
Software Update: Check
Power: Shut Down / Reboot
Configuration: Backup

Figure 2: The system lets you integrate several cameras into the overview. MotionEyeOS automatically records when it detects motion.



ON Movies

Movie File Name: %Y-%m-%d/%H-%M-%S
Movie Format: H.264/OMX (.mp4)
Movie Quality: 0% 25% 50% 75% 100%
Recording Mode: Motion Triggered
Maximum Movie Length: 0 seconds
Preserve Movies: Forever

ON Motion Detection

Frame Change Threshold: 0% 5% 10% 15% 20%
Auto Noise Detection: ON
Light Switch Detection: 0% 25% 50% 75% 100%
Despeckle Filter: OFF

Figure 3: MotionEyeOS supports multiple video formats and lets you customize the recording setup to suit your needs.

side where you can configure the camera settings. If you have integrated several cameras into the system, select the desired model in the drop-down at the top to assign settings for that specific camera.

In the *Preferences* group you first need to define how many columns of camera images you want the front end to display in the overview. Use the slide control to change the number of images arranged next to each other. This setting is useful when you want to connect multiple cameras to your Raspberry Pi and display all the images on the screen.

If you are using a low-resolution display, be careful not to activate too many columns; otherwise, the camera images will not be clear because of the extreme reduction in size. In the *General Settings* area, you can configure basic authentication, provided you have administrative privileges. The data for the administrator and viewers can be adjusted separately.

Other groups in the left pane contain settings for the specific camera. Be sure to activate the *Movies* option by switching the slide control to its left to *ON* (Figure 3). If you keep the default *OFF* setting, the software will not record any video sequences.

During activation, you also have the option of specifying in a selection box the retention period for the recordings. By default, motionEyeOS never deletes recordings, which can lead to storage space problems in the long term. That said, motionEyeOS only records video sequences and stores them on the microSD card when the program detects motion in front of the camera. Changes to the settings only take effect after pressing the *Apply* button that appears at the top right in the camera pane.

The *Working Schedule* section lets you switch video surveillance on or off on a daily basis. As soon as you use the slide control to enable the system for a specific day of the week, two additional selection fields open where you can enter times.

At the same time, a selection field appears in the lower part of the window. You can use this to enable or disable motion detection. Motion detection ensures that video streams are not displayed on the monitor without motion in the image.

Assuming you have defined a meaningful retention period for your video recordings, the software ensures that a medium-sized microSD card is all you need as a storage medium for the videos. MotionEyeOS saves the videos in the `/data/output/` directory and creates a separate subdirectory for each installed camera.

The settings dialog also has an option to change the directory for the recordings so that the system automatically stores sequences, for example, on the second partition that was created by the operating system when first launched.

Advanced

Under *General Settings*, you will see an *Advanced Settings* slide control that opens a camera-specific advanced settings dialog for highly granular configuration, such as camera resolution. This section also has basic options for the network and various services such as SSH, FTP, and Samba, which you can enable to store your video recordings on the intranet.

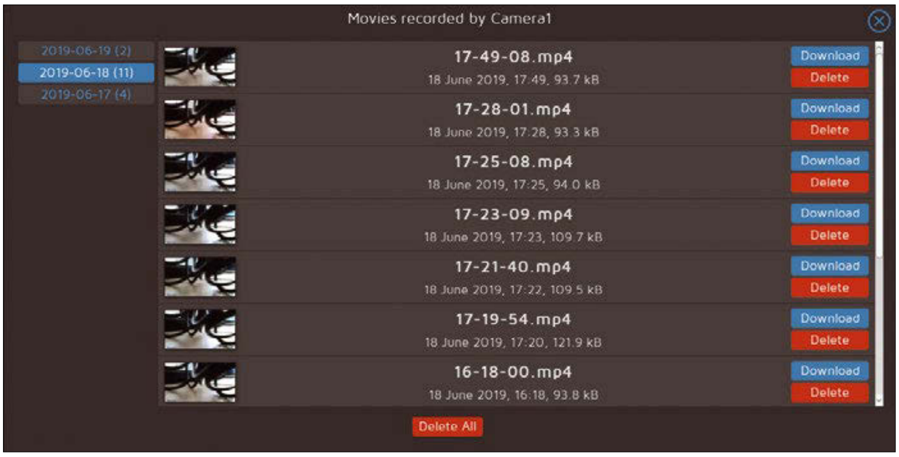
If you use a Raspberry Pi with a built-in WiFi module, it can be connected wirelessly at this point to the local network by enabling the WiFi module. In the other dialogs, which previously only allowed basic settings, you will now find various additional options. For example, the *Movies* group lets you define the quality at which recordings are saved (Figure 3); the format can be defined separately here, as well.

In the *Motion Detection* section, you can configure various settings for the built-in motion detection feature that compares images and fires up the recording function as soon as it detects changes. In this area, you define the extent to which the image must change to trigger recording, and you can even tell the system to respond to changes in lighting conditions.

When certain motion patterns are detected, the *Motion Notifications* group lets you execute defined actions, such as sending email to an address stored on the system and running a defined command. In this way, motionEyeOS notifies you when it senses unusual events, and the system can respond automatically (e.g., by shutting down devices to protect data against spying).

Playback

The videos recorded by motionEyeOS can be downloaded from the Raspberry



Movies recorded by Camera1

2019-06-19 (2)
2019-06-18 (11)
2019-06-17 (4)

17-49-08.mp4
18 June 2019, 17:49, 93.7 kB

17-28-01.mp4
18 June 2019, 17:28, 93.3 kB

17-25-08.mp4
18 June 2019, 17:25, 94.0 kB

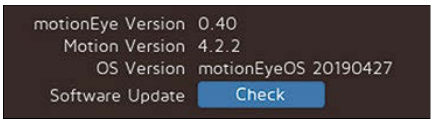
17-23-09.mp4
18 June 2019, 17:23, 109.7 kB

17-21-40.mp4
18 June 2019, 17:22, 109.5 kB

17-19-54.mp4
18 June 2019, 17:20, 121.9 kB

16-18-00.mp4
18 June 2019, 16:18, 93.8 kB

Figure 4: The images created by the software can be downloaded directly to your computer from the web front end.



motionEye Version 0.40
Motion Version 4.2.2
OS Version motionEyeOS 20190427
Software Update

Figure 5: An integrated update routine keeps the software up to date.

Pi's memory card in a dialog on the system's web front end. In the default setting, the system adds the time and duration of the recording in the file name, along with current date, so that the most important data can be traced immediately.

To open the dialog, click on the desired camera in the monitoring display on the right side of the browser window. The software then displays a horizontal button bar in the camera image at the top of the window, where you can press the *open movies browser* button. An overlapping window then displays the available recordings sorted by day, with *Download* and *Delete* buttons to the right of the respective data (Figure 4).

If you find a large number of very short recordings of just a few seconds duration in the list view, you need to adjust the *Motion Detection* settings in the administration menu. The correct setting can prevent motionEyeOS from starting a new recording with every miniscule movement it detects in an image.

Network Configuration

After commissioning motionEyeOS, you should enable the WiFi option. After the initial configuration, you no longer need the wired connection. The wireless connection lets you position the management computer freely and it no longer has to be located right next to the router. This setup allows the computer to be removed from view of unauthorized persons, which significantly improves data security.

You can activate WiFi in the *Network* section by switching the *Wireless Network* slide control to *ON* and entering the SSID of the desired network and the matching authentication key in the fields that appear. Next, press the *Reboot* button in the *General Settings* section to restart the system and integrate the Rasp Pi into your infrastructure wirelessly.

The connected WiFi cameras and the Raspberry Pi should also have assigned static IP addresses, so you will have a working system immediately after any power failures and avoid device connection failures from incorrect IP addresses.

All you have to do is change the *IP Configuration* selection boxes in the *Network* group from the default DHCP address assignments to *Manual (static IP)*. Then, switch from DHCP to the static IP address configuration in the configuration dialogs of the connected terminal devices. Alternatively, you can assign static IP addresses to all nodes in the router's web front end, removing the need for any configuration on the systems themselves.

Up To Date

If you are looking to add new cameras to a system that you operate on a Video4Linux basis, you will want to update the software up front. Because motionEyeOS is under constant development, the developers not only add new driver modules, but also frequently extend the system, adding new functions and fixing bugs.

To let you update the system, the *General Settings* group contains the *Check* button for the *Software Update* option, which motionEyeOS uses to search for updates and automatically install, if necessary (Figure 5).

Backup

Configuring motionEyeOS can be quite complex, especially if you are using several cameras with different interfaces. To avoid losing the setup you have established in the event of an accident or theft of the control computer, the *General Settings* | *Configuration* | *Backup* button lets you save all the settings.

The software prompts you for the target path; by default, this is the computer on which you access the motionEyeOS computer. The software stores the configuration files there as a `tar.gz` archive (Figure 6). The *Restore* button on the motionEyeOS computer's dashboard lets you restore the configuration files later on if need be.

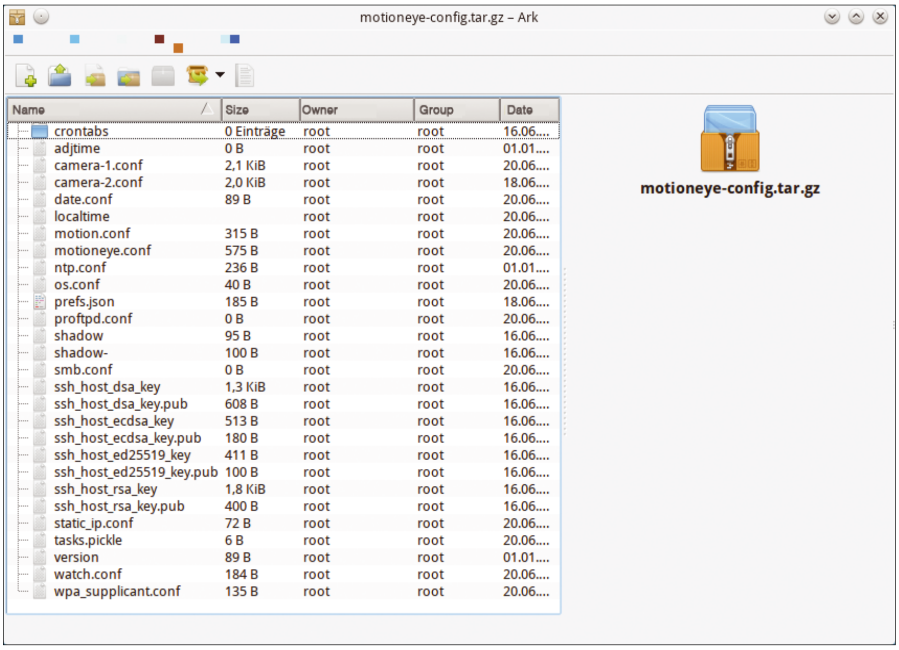
Conclusions

MotionEyeOS gives you a complete, well-thought-out, and logical video surveillance solution for your Raspberry Pi. The package also impresses with its excellent stability, great flexibility for setups that use different types of surveillance cameras, and feature set that satisfies even professional requirements. The software's intuitive operating philosophy rounds off the package. For safety-conscious users, motionEyeOS provides an excellent and efficient way to monitor exposed locations. ∎∎∎

Info

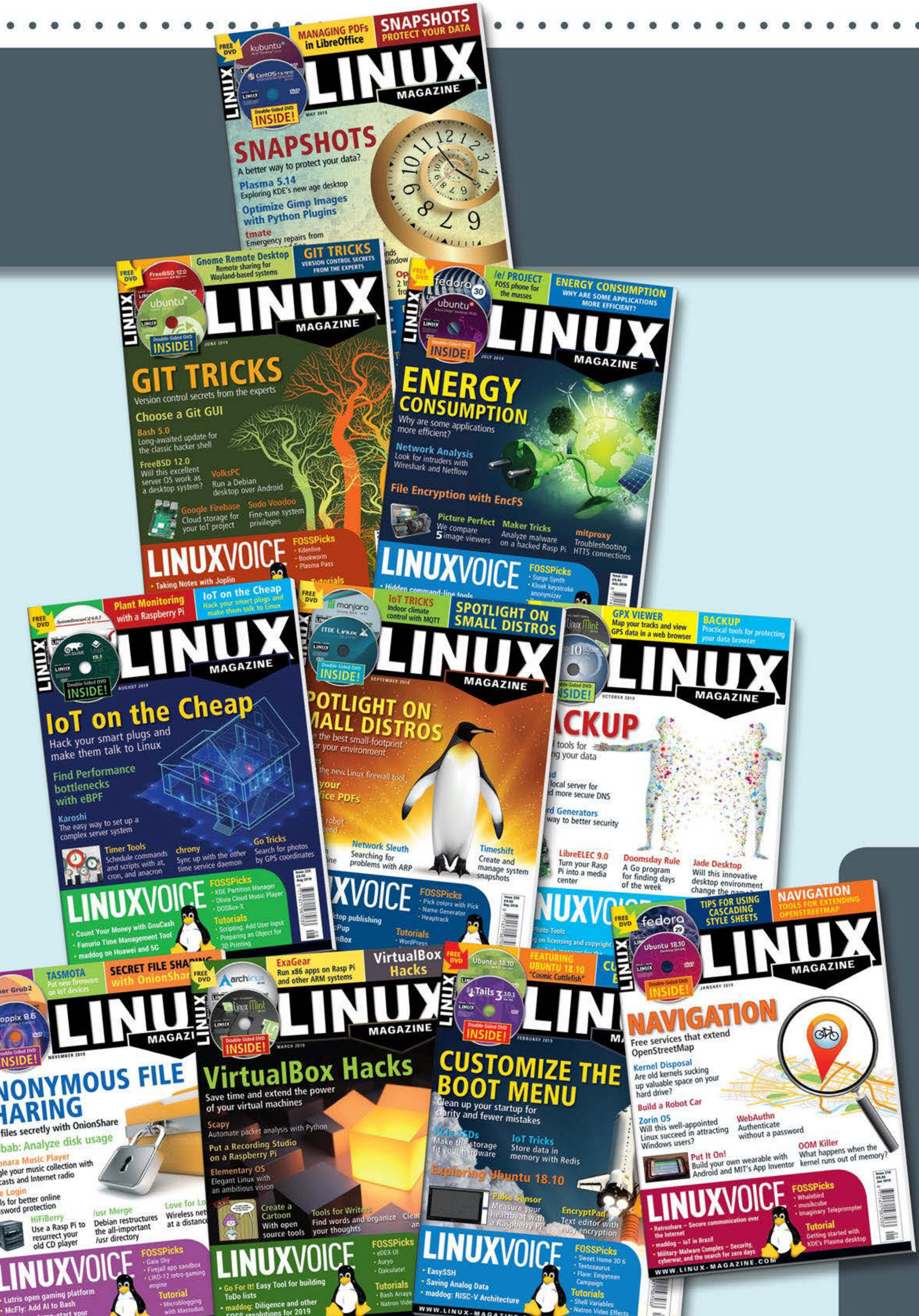
[1] motionEyeOS: *https://github.com/ccrisan/motioneyeos/wiki*

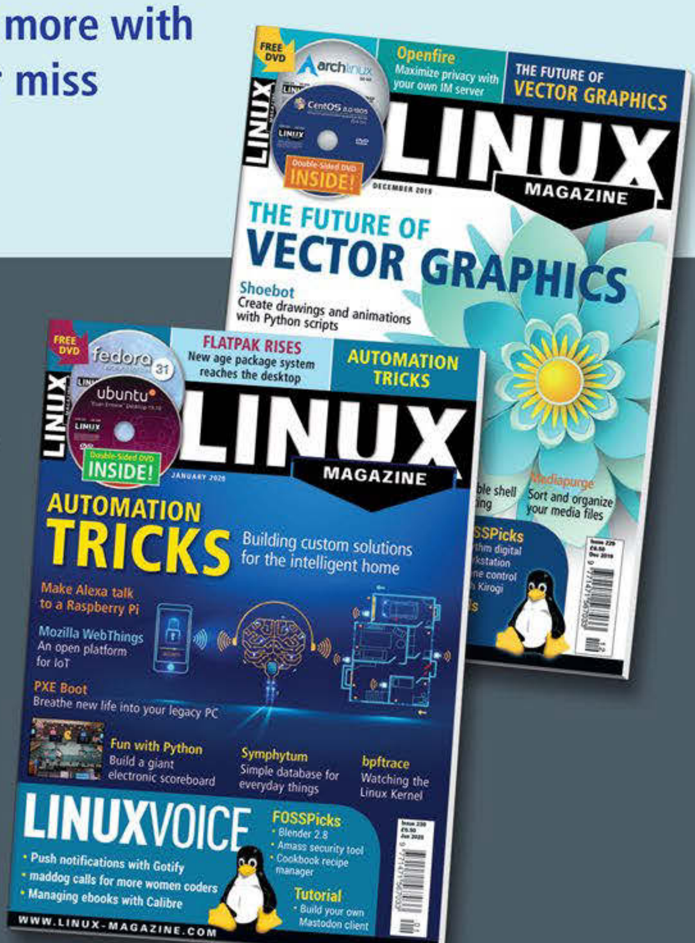
[2] Download: *https://github.com/ccrisan/motioneyeos/releases*



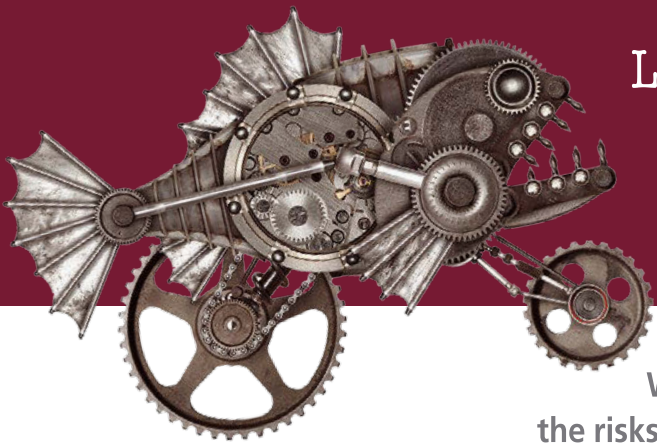
Name	Size	Owner	Group	Date
crontabs	0 Einträge	root	root	16.06....
adjtime	0 B	root	root	01.01....
camera-1.conf	2,1 KiB	root	root	20.06....
camera-2.conf	2,0 KiB	root	root	18.06....
date.conf	89 B	root	root	20.06....
localtime		root	root	20.06....
motion.conf	315 B	root	root	20.06....
motioneye.conf	575 B	root	root	20.06....
ntp.conf	236 B	root	root	01.01....
os.conf	40 B	root	root	20.06....
prefs.json	185 B	root	root	18.06....
proftpd.conf	0 B	root	root	20.06....
shadow	95 B	root	root	16.06....
shadow-	100 B	root	root	16.06....
smb.conf	0 B	root	root	20.06....
ssh_host_dsa_key	1,3 KiB	root	root	16.06....
ssh_host_dsa_key.pub	608 B	root	root	16.06....
ssh_host_ecdsa_key	513 B	root	root	16.06....
ssh_host_ecdsa_key.pub	180 B	root	root	16.06....
ssh_host_ed25519_key	411 B	root	root	16.06....
ssh_host_ed25519_key.pub	100 B	root	root	16.06....
ssh_host_rsa_key	1,8 KiB	root	root	16.06....
ssh_host_rsa_key.pub	400 B	root	root	16.06....
static_ip.conf	72 B	root	root	20.06....
tasks.pickle	6 B	root	root	20.06....
version	89 B	root	root	01.01....
watch.conf	184 B	root	root	20.06....
wpa_supplicant.conf	135 B	root	root	20.06....

Figure 6: All the critical data in a single archive: the motionEyeOS config content.


MANAGING PDFs in LibreOffice
SNAPSHOTS PROTECT YOUR DATA
LINUX MAGAZINE
MAY 2019
SNAPSHOTS
A better way to protect your data?
Plasma 5.14
Exploring KDE's new age desktop
Optimize Gimp Images with Python Plugins
tmate
Emergency repairs from
Gnome Remote Desktop
Remote sharing for Wayland-based systems
GIT TRICKS
VERSION CONTROL SECRETS FROM THE EXPERTS
JUNE 2019
GIT TRICKS
Version control secrets from the experts
Choose a Git GUI
Bash 5.0
Long-awaited update for the classic hacker shell
FreeBSD 12.0
Will this excellent server OS work as a desktop system?
VolksPC
Run a Debian desktop over Android
Google Firebase
Cloud storage for your IoT project
Sudo Voodoo
Fine-tune system privileges
LINUXVOICE
FOSSPicks
Taking Notes with Joplin
/e/ PROJECT
FOSS phone for the masses
ENERGY CONSUMPTION
WHY ARE SOME APPLICATIONS MORE EFFICIENT?
JULY 2019
ENERGY CONSUMPTION
Why are some applications more efficient?
Network Analysis
Look for intruders with Wireshark and Netflow
File Encryption with EncFS
Picture Perfect
We compare 5 image viewers
Maker Tricks
Analyze malware on a hacked Rasp Pi
mitproxy
Troubleshooting HTTS connections
LINUXVOICE
FOSSPicks
Plant Monitoring with a Raspberry Pi
IoT on the Cheap
Hack your smart plugs and make them talk to Linux
AUGUST 2019
IoT on the Cheap
Hack your smart plugs and make them talk to Linux
Find Performance bottlenecks with eBPF
Karoshi
The easy way to set up a complex server system
Timer Tools
Schedule commands and scripts with at, cron, and anacron
chrony
Sync up with the other time service daemon
Go Tricks
Search for photos by GPS coordinates
LINUXVOICE
FOSSPicks
Count Your Money with GnuCash
Fanurio Time Management Tool
maddog on Huawei and 5G
IoT TRICKS
Indoor climate control with MQTT
SPOTLIGHT ON SMALL DISTROS
SEPTEMBER 2019
SPOTLIGHT ON SMALL DISTROS
Network Sleuth
Searching for problems with ARP
Timeshift
Create and manage system snapshots
GPX VIEWER
Map your tracks and view GPS data in a web browser
BACKUP
Practical tools for protecting your data browser
OCTOBER 2019
BACKUP
LibreELEC 9.0
Turn your Rasp Pi into a media center
Doomsday Rule
A Go program for finding days of the week
Jade Desktop
Will this innovative desktop environment change the game?
TASMOTA
Put new firmware on IoT devices
SECRET FILE SHARING with OnionShare
NOVEMBER 2019
ANONYMOUS FILE SHARING
Share files secretly with OnionShare
Baobab: Analyze disk usage
HiFiBerry
Use a Rasp Pi to resurrect your old CD player
/usr Merge
Debian restructures the all-important /usr directory
LINUXVOICE
FOSSPicks
Lutris open gaming platform
ExaGear
Run x86 apps on Rasp Pi and other ARM systems
VirtualBox Hacks
MARCH 2019
VirtualBox Hacks
Save time and extend the power of your virtual machines
Scapy
Automate packet analysis with Python
Put a Recording Studio on a Raspberry Pi
Elementary OS
Elegant Linux with an ambitious vision
Create a Cartoon
With open source tools
Tools for Writers
Find words and organize your thoughts
LINUXVOICE
FOSSPicks
Tutorials
Go For It! Easy Tool for building ToDo lists
maddog: Diligence and other FOSS resolutions for 2019
FEATURING UBUNTU 18.10 "Cosmic Cuttlefish"
FEBRUARY 2019
CUSTOMIZE THE BOOT MENU
Clean up your startup for clarity and fewer mistakes
PCIe SSDs
Make the storage fit your hardware
IoT Tricks
Store data in memory with Redis
Exploring Ubuntu 18.10
Pulse Sensor
Measure your heartbeat with a Raspberry Pi
EncryptPad
Text editor with easy encryption
LINUXVOICE
FOSSPicks
Tutorials
EasySSH
Saving Analog Data
maddog: RISC-V Architecture
TIPS FOR USING CASCADING STYLE SHEETS
NAVIGATION
TOOLS FOR EXTENDING OPENSTREETMAP
JANUARY 2019
NAVIGATION
Free services that extend OpenStreetMap
Kernel Disposal
Are old kernels sucking up valuable space on your hard drive?
Build a Robot Car
Zorin OS
Will this well-appointed Linux succeed in attracting Windows users?
WebAuthn
Authenticate without a password
Put It On!
Build your own wearable with Android and MIT's App Inventor
OOM Killer
What happens when the kernel runs out of memory?
LINUXVOICE
FOSSPicks
Tutorial
Getting started with KDE's Plasma desktop
Retroshare – Secure communication over the Internet
maddog – IoT in Brazil
Military-Malware Complex – Security, cyberwar, and the search for zero days
WWW.LINUX-MAGAZINE.COM


LINUX MAGAZINE
THE FUTURE OF VECTOR GRAPHICS
Openfire
Shoebot
LINUX MAGAZINE
AUTOMATION TRICKS
FLATPAK RISES
LINUXVOICE
WWW.LINUX-MAGAZINE.COM

MakerSpace



Librem 5: The risks of crowdfunding open hardware

Risky Business

With the Librem 5 as a cautionary tale, Bruce ponders the risks of backing crowdfunding campaigns. *By Bruce Byfield*

Librem 5 [1] (Figure 1), Purism's free-licensed phone, was supposed to be a major milestone in open hardware. Other efforts at a free phone, ranging from Openmoko to the Ubuntu Touch, have failed in the past, but the Librem 5 looked like it might be different. Yet as the Librem 5 starts to ship, some are already taking its difficulties as proof that a free phone is unpractical. Very likely, the problems will eventually be fixed, but, for now, the Librem 5 is starting to look like a cautionary tale about the dangers of crowdfunding and of backing unproven open hardware.

Figure 1: The long-anticipated release of the Librem 5 phone has yet to live up to expectations.

In August 2017, the Librem 5 was announced with a crowdfunding campaign. Already a critical success with a line of free laptops, Purism created considerable buzz with a series of announcements about partnerships with Gnome, KDE, and other projects in the Librem 5's development. Cynics voiced doubts, but many in the media – including me – expressed enthusiastic and repeated support. After all, the Librem 5 was something that many advocates of open source had awaited for years. It helped, too, that Purism brushed aside the difficulties of developing a phone that depended on neither Android nor iOS. In a 2018 interview with *Linux Pro Magazine*, Todd Weaver, Purism's CEO, said simply, "this is a simple equation of momentum and time. We have hundreds of development kits shipping to active developers. Plus, we have emulators, so we will see more and more applications being written and ported to PureOS. It is only a matter of time."

However, as the shipping date grew near, problems started to arise. Originally scheduled for April 2019, ship-

Lead Image © Burmakin Andrey, 123RF.com

ping did not begin until September 2019 – and then only to Purism staff and selected developers. Such delays are not unusual in any product manufacturing, but rumors began to leak out from ex-employees that Purism faced financial problems and was internally in chaos. To make matters worse, Librem 1, a suite of services for the phone, proved mediocre and inconsistent in design, and reports of overheating circulated.

Finally, in late November, the first customers began to receive their phones. However, as Jim Salter of Ars Technica reported, relating the early experience of a backer called Azdle, what was shipped was more of a prototype than a functional phone [2]:

"First of all, it's not really a 'phone' yet. There's no audio when attempting to place a phone call. The cameras also don't appear to work yet. Azdle reports 'installing and opening up Cheese' – Cheese is a very basic Linux video application, installed by default in many distros – 'I just get a message saying "no device found".' There's also effectively no power management yet, so the Librem doesn't last long on battery. It takes a long time to charge as well.

"The software needs polish in lots of places: Azdle notes that few apps so far understand mobile screen layouts, and there's no obvious indications as to which apps have or have not been updated. The charging LED doesn't light up when the phone's off – although the phone is actually charging. And fine-tunables like kinetic scroll – the ability to flick a scroll-thumb down hard, and expect it to keep scrolling for pages and pages like a thrown rock – still need tuning."

In other words, what is shipping as I write is lacking two of the main functions expected of a modern phone and is rough around the edges generally. With any luck, software upgrades will gradually provide the expected functionality, but could anything be more disappointing after two years of anticipation?

As I write, any financial difficulties appear to have been staved off by a second round of investors, but supporters might be forgiven for having briefly worried whether they would ever receive a fully functioning product. At the very least, supporters might wonder what would happen if they backed another project that depended on sales alone to deliver the promised hardware.

Why Transparency Matters

Such worries are not unfounded. Crowdfunding campaigns that do not deliver what they promise have happened before. In 2016, Kickstarter suspended the Skarp Laser Razor campaign [3], apparently due to doubts that the product could be delivered. Similarly, in 2012, Eyez, a Google Glass-like product [4], never shipped. Admittedly, the Librem 5 is not yet in this infamous company, but the apprehensions that it might be should force the crowdfunding campaign backers to think twice about what they are doing. The caution is especially needed with open hardware, which is the center of so many hopes. Just to look through open hardware campaigns is to fall into a hardcore techie's version of SkyMall or the Lee Valley Tools catalog and to be lost in a dazzling array of gadgets. In the excitement, it can be hard to soberly assess a campaign or to estimate its chances of delivery.

In the case of the Librem 5, backing the project originally seemed like a safe gamble. Purism is incorporated as a social purpose company, which means it can make decisions based on social concerns rather than being focused entirely on profit [5]. Even more importantly, it has applied for Respects Your Freedom certification for its laptops, and speakers for the company have regularly voiced support for the privacy positions of the Electronic Frontier Foundation and the Free Software Foundation. To open source advocates like me, Purism appears to be one of us and trustworthy. And in fact, this same line of reasoning is the main reason to believe that Purism will eventually release a fully-functional Librem 5.

But what if Purism, or some other company in the same position, collapses financially and is unable to deliver on its promises, even with the best of intentions? How can backers foresee that event? Certainty does not exist, and backing any crowdfunding campaign means the risk of losing your money. This is a seemingly obvious fact, although no warnings of it appear to be on any of the crowdfunding sites.

No simple answer exists. However, in looking at how Purism has handled the crisis of its releases, what strikes me is the lack of transparency in its public statements, especially when compared to other crowdfunded open hardware projects. Purism has explained next to nothing of its circumstances, stating only how it plans to manage the release – and the resulting vacuum has filled quickly with rumors.

By contrast, the EOMA68 recyclable computer (Figure 2), which I wrote about several years ago, has yet to ship at all. My impression is that project head Luke Leighton severely underestimated the amount of work required to bring a product to market. Feature creep may have also occurred. Yet the project struggles on. Leighton has repeatedly apologized for the delay and continues to write updates every couple of months, explaining what has

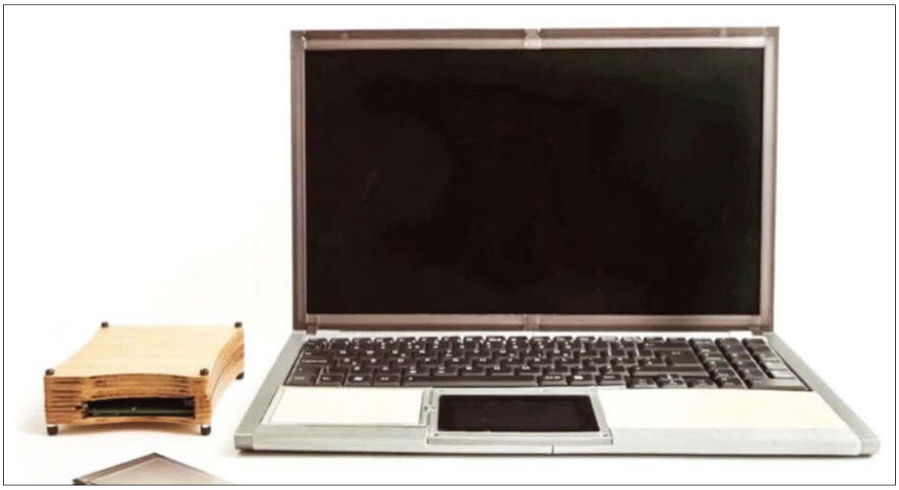


Figure 2: The EOMA68 recyclable computer has yet to ship, but backers are sustained by regular progress updates.

caused delays and what his current efforts are. In his November 2019 update [6], Leighton writes:

> *"I'm grateful to those people who get it and say 'thank you' for the level of detail and the description of the experiences that we've had, because for* their project, *they then know what to expect, and enjoy being educated on the kinds of pitfalls that might occur, and how to get out of them. I learned from the OpenPandora and the Openmoko and many other projects, and am repaying that in kind."*

Leighton's transparency helps to sustain his backers' faith in him. He may be learning as he goes, but at least some still believe that he is trying to deliver on his promises. Just as importantly from his point of view, because he keeps his backers informed, there is less room for rumors. By chronicling his mistakes, he may help future open hardware projects to achieve their goals. In that way, even if he fails, he is doing a service to the community.

Much the same is true of Keyboardio, which manufactures the Model 01 keyboard. Keyboardio took nearly three years to deliver its keyboard after its successful crowdfunding campaign, yet while backers chafed, they mostly remained patient. Throughout that time, Keyboardio's Jesse Vincent and Kaia Dekker demonstrated their prototypes extensively and wrote detailed blogs [7] about the challenges of development and manufacturing and of doing business in China (Figure 3). As with Leighton, no one could doubt their intentions. Keyboardio delivered their product two years ago; they are now developing new products and have transformed Keyboardio into a small business.

The lesson of EOMA68 and Keyboardio is that manufacturers seem more reliable when they communicate freely about what they are doing. Transparency is a mark of sincerity and helps backers to trust the crowdfunding project. Purism's misstep seems to be that they have failed to keep their backers informed. Obviously, it would be embarrassing to admit to financial difficulties, if any existed, yet failing to communicate only makes their backers fear the worst. Of course, transparency is no guarantee of success, but it seems likely that any open hardware manufacturer who goes to the trouble of being transparent is also likely to be conscientious enough to deliver what they promised.

Playing the Odds

Transparency is not the only way to decide whether to back a project. Increasingly, I find myself making decisions about whether to back a project depending on the site that hosts the campaign. My personal preference is for Crowd Supply. Compared to Kickstarter and Indiegogo, Crowd Supply is a small site, but it works closely with campaigns to refine their approach and also sells products once they are released. By my count, its campaigns have an 85 percent success rate, 12 percent higher than its rivals. Because Crowd Supply is vested in the success of the projects it hosts, its projects seem a safer bet.

Similarly, I am steering clear of overly ambitious campaigns. For instance, I would welcome a Linux tablet, but having reported on Aaron Seigo's failed Vivaldi project, I would hesitate to back a similar effort, because I have seen the challenges. I would be more likely to invest in campaigns that involved writing software and using preexisting parts as much as possible.

I hope that Purism prevails through its present difficulties. The hopes for the Librem 5 were so high in the community that its failure would be devastating. Still, as the community waits for events to unfold, the situation can serve as a warning to be cautious about supporting open hardware projects. The excitement of innovation should not make backers forget that they are taking a risk, much like a venture capitalist. At the very least, some due diligence seems necessary. ■■■

Figure 3: Transparency and public appearances kept Keyboardio backers patient despite manufacturing delays and setbacks.

Info

[1] Librem 5: *https://shop.puri.sm/shop/librem-5/*

[2] Early report on the Librem 5: *https://arstechnica.com/gadgets/2019/12/librem-5-backers-receiving-their-linux-phones/*

[3] Skarp Laser Razor campaign: *https://www.kickstarter.com/projects/skarp/the-skarp-laser-razor-21st-century-shaving/*

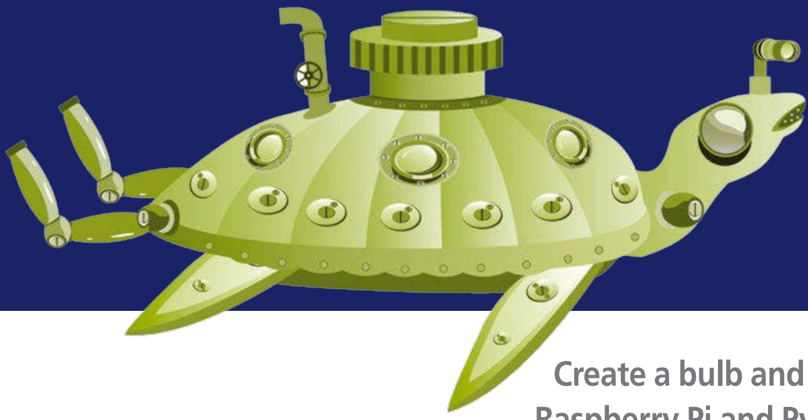
[4] Eyez campaign: *https://www.kickstarter.com/projects/zioneyez/eyeztm-by-zioneyez-hd-video-recording-glasses-for*

[5] Social purpose corporation: *https://puri.sm/about/*

[6] Crowd supply EOMA68 Update, "Measurements and a Hypothesis," November 30, 2019: *https://www.crowdsupply.com/eoma68/micro-desktop/updates/measurements-and-a-hypothesis*

[7] Keyboardio blog: *https://blog.keyboard.io/*

MakerSpace



Programming with Python Turtle graphics

Turtles

Create a bulb and tube analog thermometer with a Raspberry Pi and Python Turtle graphics. *By Brooke and Pete*

Turtle graphics are a great way to get kids started programming. The Python *turtle* library offers simple step-by-step graphical methods that allow young programmers to create graphics with only a few lines of code.

I found that once my kids understood Turtle basics, we were able to do more advanced Raspberry Pi projects that used sensors, push buttons, and outputs. In this article, I share a Rasp Pi Turtle project that creates a graphic of an "old style" mercury thermometer animated with a Pi temperature sensor.

Getting Started

The Python *turtle* library is probably already loaded on your Raspberry Pi system, but if it is not, enter:

```
pip install turtles
```

The *turtle* library [1] is used a little bit as you would use a pen on a drawing board. With the use of methods (e.g., `forward()`, `backward()`, `left()`, and `right()`), you can draw lines. If you want to move a turtle without drawing a line, you use the `penup()` method; then, once you arrive at the desired location, the `pendown()` method lets you

Listing 1: turtle1.py

```
from turtle import *

Screen()
title(" 3 Turtles")

# The first turtle creates an open rectangle
t1=Turtle() # new turtle at (0,0) facing right
t1.shape("turtle")
t1.left(90) # Rotate left 90 degree. This will face up
t1.forward(200) # move 200 straight
t1.left(90) # Rotate left 90 degree. This will face left
t1.forward(50)
t1.left(90) # Rotate left 90 degree. This will face down
t1.forward(200)

# The second turtle is red.
# It moves to a new position and draws a thick circle
t2=Turtle()
t2.color("red")
t2.pensize(10) # Draw a thick line
t2.up()
t2.goto(-25,-120)
t2.down()
t2.circle(50) # Draw a circle with a radius of 50

# The third turtle writes some blue text
t3=Turtle()
t3.color("blue")
t3.penup()
t3.goto(30,75)
t3.write("My first turtle app") # Put text on the screen
```

Lead Image © Baurzhan Moldabekov, 123RF.com

start drawing again. Turtle objects have properties for color and line thickness and a number of additional methods for shapes, fill, circles, and text.

To create a turtle object, you use the `myturtle = Turtle()` statement. A new turtle is placed at the center of the screen, position (0,0), facing right.

The Python Turtle example in Listing 1 shows three basic concepts I use in this project: drawing lines, moving without drawing, and adding text (Figure 1).

To draw `t1`, named `turtle` (line 7), the turtle (pen) rotates 90 degrees left to face toward the top of the drawing area (line 9). With a combination of `left` and `forward` methods, turtle `t1` becomes an open rectangle (lines 9-14).

The `t2` shape is red and drawn with a thick pen size (lines 18-20). The `up()` method makes sure the pen does not draw while it moves (line 21-22). After reaching the new position, the pen goes back down to draw a circle (lines 23-24). The pen for turtle `t3` goes to a new position without drawing a line and writes some text on the screen (line 27-31).

Once you (or your young programmer) have mastered drawing lines, moving without drawing, and adding text, you can start some more interesting projects.

Drawing a Thermometer

For this project, I wanted to draw an "old style" mercury thermometer with a bulb of red mercury at the bottom and a tube above it (Figure 2).

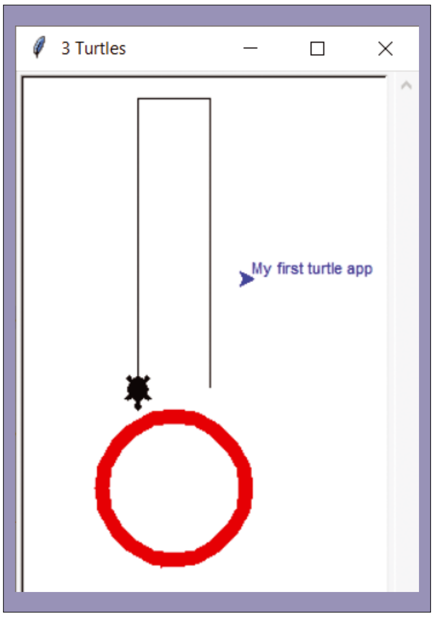
Simple turtle commands like `move`, `left`, `right`, and `forward` are great for drawing simple graphics, but they can be awkward if you want to create a more complex drawing. A more efficient approach is to define an array of (*x*, *y*) coordinates and move to each position in the array. For example, the upper tube can be drawn with:

```
# An array of coordinates for the tube
outline = ((25,-50),(25,210),(-25,210),➥
           (-25,-50))
for pos in outline: ➥
   # move to each tube x,y point
   t1.goto(pos)
```

To draw the lower "mercury" bulb in this project, I enhance the circle of my first example, turtle `t2`, by creating a black circle with red fill. The fill is toggled with the `begin_fill()/end_fill()` methods:

```
# put the pen up and move to the circle ➥
  starting point
t2.penup()
t2.goto(0,-137)
t2.pendown()
t2.pensize(5)
t2.color("black","red")
# draw the circle with fill
t2.begin_fill()
t2.circle(50)
t2.end_fill()
```

At this point, I've created a static background for the thermometer. The next step is to read a temperature value from the Raspberry Pi and add dynamic information to the drawing.



My first turtle app

Figure 1: Three Python Turtle graphics.

Hardware Setup

A number of different temperature sensors can be used for this project. Here, I use a low-cost ($5) DHT11 temperature/humidity sensor [2]. This sensor has three pins, (signal, 5V, and GND; Figure 3). The signal pin connects to Rasp Pi physical pin 7 (PP7). The program in Listing 2 tests the DHT11 sensor. Note that because the sensor can return a humidity value, you could add that information in your Turtle code, as well (line 12).

To install the DHT temperature sensor Python library [3], enter:

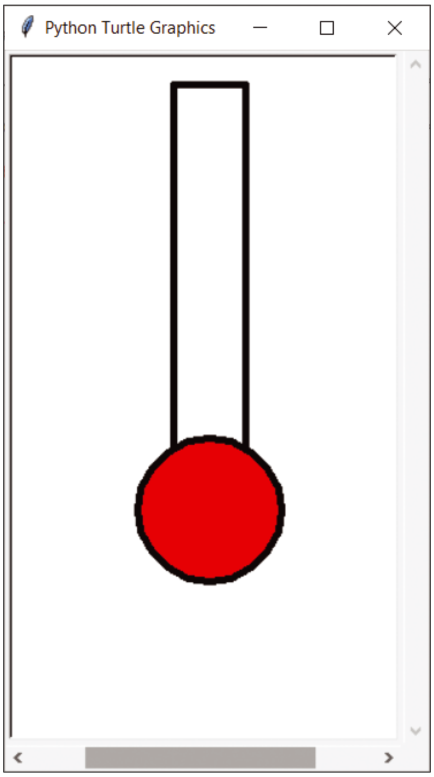


Figure 2: Thermometer background.

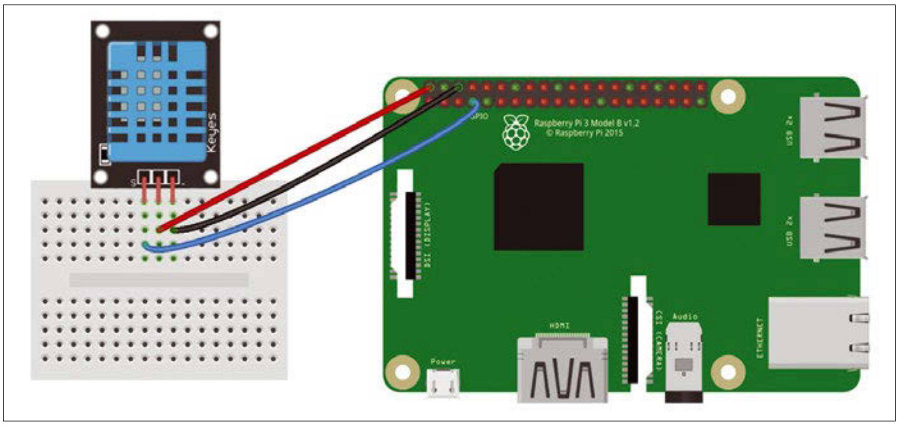


Figure 3: Fritzing diagram of DHT11 wiring to the Rasp Pi; the blue wire is signal (PP7), the red wire is 5V power (PP2), and the black wire is ground (GND, PP6).

Listing 2: test.py

```
#!/usr/bin/python
import sys
import Adafruit_DHT

sensor_type = 11 # sensor type could also be 22, for DHT22

dht_pin = 4 # Note: BCM pin 4 = physical pin 7

humidity, temperature = Adafruit_DHT.read_retry(sensor_type, dht_pin)

print( "Temp: ", temperature, " deg C")
print( "humidity: ", humidity, " %")
```

```
sudo pip install Adafruit_DHT
```

When I'm doing Pi projects with my kids, I like to use prototyping expansion tops (Figure 4). These tops start at around $6 [4]. Not only do they make the wiring cleaner, it's a lot easier to move the projects around for testing.

Turtle Thermometer

For the thermometer project (Listing 3), I use two turtles: a static background turtle (`thermo`, line 27) and a dynamic turtle (`bar`, line 29). The background turtle is used only once at the start of the program. The drawing and text from the dynamic turtle (`bar`) are erased with the `bar.clear` method (line 8) and then redrawn with each new temperature value.

The final turtle thermometer code is a little more complex than the earlier code, because I include temperature labels (lines 42-46) and a `drawbar()` function (lines 6-19) that gets a new temperature value and redraws the dynamic bar and the text that indicates the current value (Figure 5).

A screen object `wn` (line 22) resizes the window, adds a title, and defines an `ontimer` function (lines 19 and 58). The `wn.ontimer(drawbar, 5000)` statement updates the thermometer graphic every five seconds.

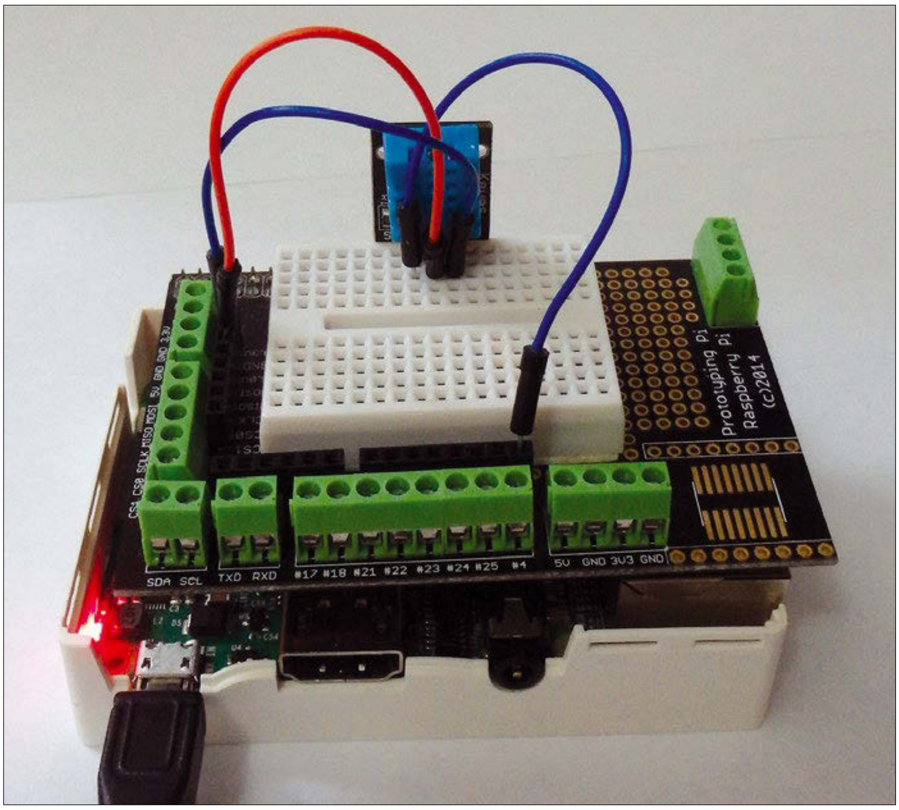


Figure 4: DHT11 wiring to the Rasp Pi through a prototyping board.

Final Comments

The mercury thermometer project was a fun example of how to use some basic Python Turtle code to talk to Raspberry Pi inputs. To include user input in your projects, the *turtle* library has an `onkey(mykeyfunc,key)` method to listen for key presses and an `onscreenclick(onclick_handler)` call to listen for mouse clicks.

Compared with other Python graphics libraries (e.g., Pygame, Tkinter, or Qt), Turtle graphics are slow and perhaps limiting, but for kids projects, Turtle can be a nice, easy introduction to graphics programming.

Raspberry Pi Turtle projects can be fun. Some other projects my daughters and I have created include:

- Drawing pictures with a Wii remote
- Showing the path of a rover while it is being driven
- Creating a "Spirograph" drawing by entering colors and turn angles ■■■

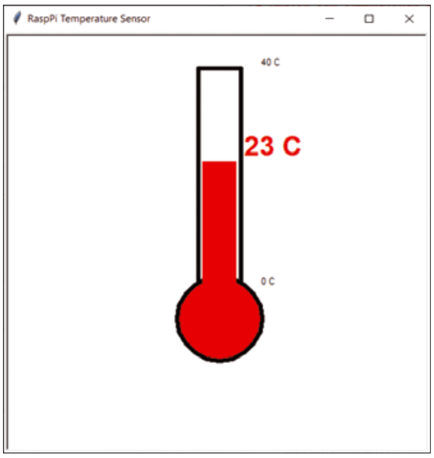


Figure 5: Rasp Pi Turtle thermometer.

Info

[1] Python Turtle graphics: *https://docs.python.org/3.3/library/turtle.html?highlight=turtle*

[2] DHT11 sensors: *https://www.adafruit.com/product/386*

[3] DHT Python library: *https://github.com/adafruit/Adafruit_Python_DHT*

[4] Pi prototyping top: *https://www.dx.com/p/diy-prototyping-expansion-board-for-raspberry-pi-b-red-2047708*

Author

You can investigate more neat projects by Pete Metcalfe and his daughters at *https://funprojects.blog*.

Listing 3: turtle2.py

```
from turtle import Turtle,Screen, mainloop
import random, time
import Adafruit_DHT

# Update the temperature bar height and value
def drawbar():
       temp = random.randint(5,40)
       bar.clear()  # clear the old bar and text
       top = (-50 + 260 * temp/40)
       boutline = ((0,-50),(20,-50),(20,top),(-20,
                   top),(-20,-50),(0,-50))
       bar.begin_fill()
       for pos in boutline:
              bar.goto(pos)
              bar.down()
       bar.end_fill()
       bar.up()
       bar.goto(30,top)
       bar.write(str(temp) + " C",font=("Arial",24,
                                        "bold"))
       wn.ontimer(drawbar, 5000)

# Setup a default screen size and Title
wn = Screen()
wn.setup(width = 500, height = 500)
wn.title("RaspPi Temperature Sensor")

# define a static thermo backgroup object and a dynamic
   bar object
thermo = Turtle()
thermo.penup()
bar = Turtle()
bar.color("red")
bar.hideturtle()

# define an array for the top tube
outline = ((25,-50),(25,210),(-25,210),(-25,-50))
thermo.hideturtle()
thermo.pensize(5)
for pos in outline:
       thermo.goto(pos)
       thermo.pendown()

# add some temperature labels
thermo.penup()
thermo.goto(50,-50)
thermo.write("0 C")
thermo.goto(50,210)
thermo.write("40 C")

# draw the filled bulb at the bottom
thermo.goto(0,-137)
thermo.pendown()
thermo.pensize(5)
thermo.color("black","red")
thermo.begin_fill()
thermo.circle(50)
thermo.end_fill()

# setup an ontimer function
wn.ontimer(drawbar, 5000)
wn.mainloop
```

IT Highlights at a Glance

The New IT
ADMIN – your source for technical solutions to real-world problems.
SUBSCRIBE NOW!

HPC Up Close

- OpenACC – Data Management
- Researchers Introduce Deep500 Benchmark
- US Government Shutdown Ties Up $139.2 Million in Grant Funding
- Packet Analysis with Scapy

Further Reading

- GUI or Text-Based Interface?
- Shared Storage with NFS and SSHFS
- Intel Unveils New Processor Line
- Quantum Computing Milestone Achieved
- DES Completes Dark Matter Survey

Highlights

OpenACC – Data Management
The OpenACC data directive and its associated clauses allow you to control the movement of data between the host CPU and ...PU.

...troduce Deep500 Benchmark
...s detailed analysis of deep-learning ...C systems

...t Shutdown Ties Up $139.2 Million ...ng
... for forbearance and vows to start processing

...s with Scapy
...ed up Wireshark a time or two to ...plication or system problem. During ...ptures (PCAPs) are...

★ Complete Your Library ★
Order a Full Year of Digital Archives and Save at Least 50%!
2018 AVAILABLE NOW!

Special Offer from ADMIN

...ork & Security magazine is your source for ...tions to real-world problems.

CLOUD EXPO EUROPE
REGISTER FOR YOUR FREE TICKET TODAY!
12-13 March 2019
ExCeL London

ADMIN Update – Hottest Links

- Migration in the Cloud
- A 19-Year-Old Bug in WinRAR
- Hackers Started Exploiting Drupal Bug
- Rook
- Chef

Further Reading

- LibreOffice Vulnerable to Remote Code Execution Flaw
- Kernel Bonding vs. libteam
- PowerShell Best Practices
- Checking USB Performance
- Keep All Your Linux Servers in Check

Highlights

Migration in the Cloud
Move a workload to the cloud without trouble, and leverage cl... benefits for a conventional setup. (more)

A 19-Year-Old Bug in WinRAR
Popular Windows archive tool is vulnerable to attack. (more)

Hackers Started Exploiting Drupal Bug
The bug affects Drupal 8 sites. (more)

Rook
Kubernetes is known to be inflexible when it comes to persis... storage, but Rook now offers cloud-native storage and seeks ... achieve harmony. (more)

Chef
Chef, one of the oldest automation solutions, seeks to become ... universal administration robot for cloud environments with ne... version 2.0. (more)

Most Read Articles

Pen Testing
Discover indicators of compromise with open source pen testi... tools. (more)

LINUX UPDATE
EXPLORING THE WORLD OF LINUX

- Configuring Bluetooth on Linux
- Sudo Vulnerability
- openSUSE OBS Can Now Build Windows WSL Images
- Scribus
- Tutorial – Fediverse

FEATURED ARTICLES

Configuring Bluetooth on Linux
Getting Bluetooth and PulseAudio to play nicely takes a little work. Bruce walks you through the basics of getting Bluetooth up and running on Linux. (more)

Sudo Vulnerability
A vulnerability in the sudo package gives sudo users more powers than they deserve. (more)

openSUSE OBS Can Now Build Windows WSL Images
openSUSE enables developers to build their own WSL distributions. (more)

Scribus
Scribus is a desktop publishing tool for Linux that expands your options when designing simple flyers and brochures, giving you plenty of room to be creative. (more)

Tutorial – Fediverse
If you're looking for social media options where the user has more control, you'll find a range of options to explore in the Fediverse, including the popular Mastodon. (more)

FURTHER READING

- GnuCash: Bookkeeping the FOSS Way
- Red Hat Announces CentOS Stream
- Richard Stallman Resigns from Free Software Foundation
- Jump Box Security
- SystemRescueCd

Get hyped.
The International Conference for High Performance Computing, Networking, Storage, and Analysis
SC19 hpc is now.

Linux Magazine Trial

MOST READ

chkrootkit
Linux can be infected by rootkit malware that is hidden and hard to detect. The chkrootkit program can help find rootkit infections. (more)

Linus Torvalds Agrees to Kernel Lockdown

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. Subscribe today for our excellent newsletters:

ADMIN HPC • ADMIN Update • Linux Update

and keep your finger on the pulse of the IT industry.

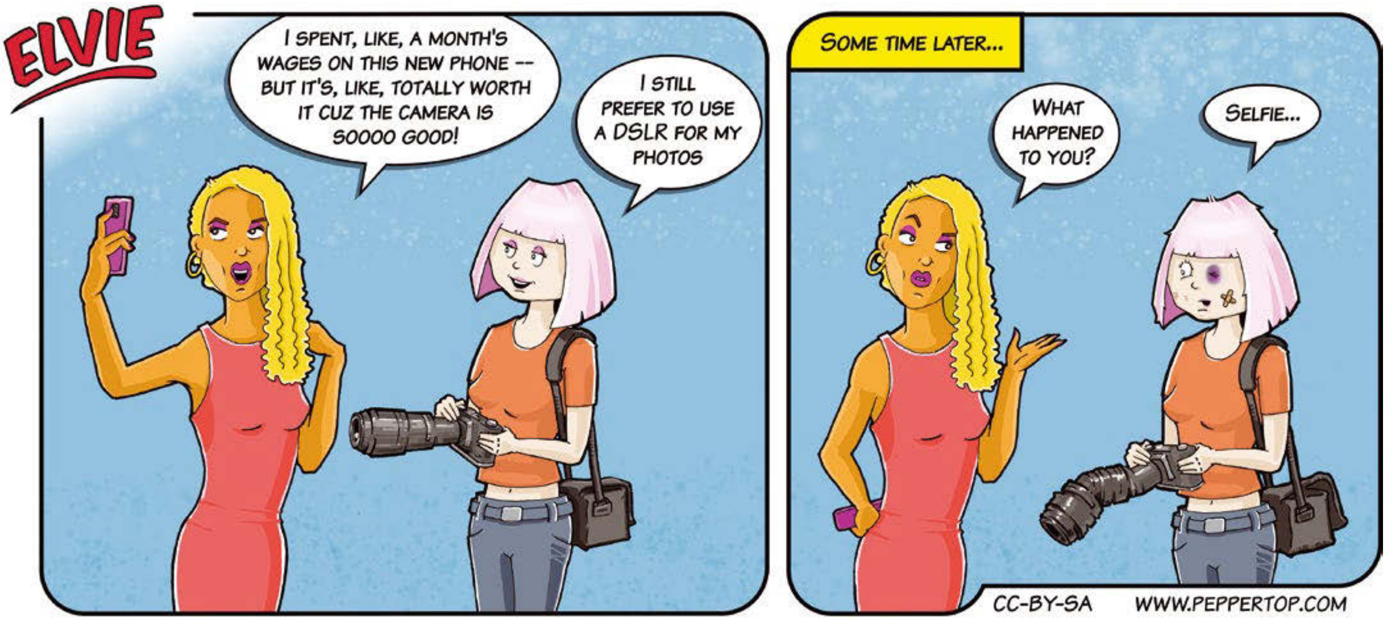
Admin and HPC:	https://bit.ly/HPC-ADMIN-Update
Linux Update:	https://bit.ly/Linux-Update

Photo by Parker Byrd on Unsplash

"Simple tools for a single purpose" is supposed to be the philosophy of Unix and the Linux revolution, but we humans are forever enticed by the goal of building one tool that will do everything. For many years, the open source world has toyed with integrating email with Personal Information Manager (PIM) features like calendaring and contacts. An innovative application called Kube takes this illusive concept a step further – not just integrating but blurring the edges between the various components. This month we dive into Kube and show you how it works and why it might be a little different from what you're using now. Also inside, we take a snapshot of the Geeqie photo manager, and we deliver another useful installment in our series on the Mastodon open source social-networking tool.

Image © Olexandr Moroz, 123RF.com

LINUXVOICE


ELVIE
I SPENT, LIKE, A MONTH'S WAGES ON THIS NEW PHONE -- BUT IT'S, LIKE, TOTALLY WORTH IT CUZ THE CAMERA IS SOOOO GOOD!
I STILL PREFER TO USE A DSLR FOR MY PHOTOS
SOME TIME LATER...
WHAT HAPPENED TO YOU?
SELFIE...
CC-BY-SA
WWW.PEPPERTOP.COM

MADDOG'S DOGHOUSE

Maddog considers the joys of recreating ancient computers and learning about programming within these older architectures. BY JON "MADDOG" HALL

Jon "maddog" Hall is an author, educator, computer scientist, and free software pioneer who has been a passionate advocate for Linux since 1994 when he first met Linus Torvalds and facilitated the port of Linux to a 64-bit system. He serves as president of Linux International®.

PiDPs and Gigatron

Recently, there was much discussion on my social media pages about ancient computers and teaching assembly language with regards to these older architectures.

I have long advocated that even in the days of virtual machines and Java that computer science students and computer engineering students should learn at least one assembly language early in their programming education. This, in my mind, is necessary to avoid mistakes in program design and to be able to discover errors in compilation that might occur due to the compiler or optimizations requested.

Perhaps some of you are not aware of the magnificent work of Bob Supnik, a vice president and engineer who I had the privilege of knowing and working with for a number of years at DEC.

Even before he left DEC, Bob started working on an emulator now called SimH [1], which runs on a variety of different operating systems and emulates an even larger number of hardware architectures. Not just DEC architectures … *many* different architectures.

Using this emulator, you can run old *binaries* of operating systems and programs on your laptop, and, armed with hardware manuals and old literature off the net, you too can program these old machines in assembly languages and run old operating systems, compilers, etc., to your heart's content.

But wait! There's more!

Those of you who know me know how much I loved two architectures: The PDP-11 for its elegance and the PDP-8 for its simplicity (and because it was the second computer I ever programmed, and the first assembly language I ever learned). However the usefulness of learning machine/assembly language is amplified by learning *both* the PDP-8 and the PDP-11 architectures, to see the difference between the two.

The PDP-8 was enormously (for its time) successful, for it could fit on a tabletop and was fast enough to process data, mostly from engineering/scientific types of problems. The PDP-8 had only 4,096 12-bit words and could not even subtract. To subtract you had to take the two's complement of the subtrahend and add it to the minuend (I have not even thought of those terms in about 20 years), and then check for overflow.

The PDP-11, the predecessor to the VAX line of computers, was probably one of DEC's most profitable and longest lasting architectures, with its refrigerator sized racks of circuit boards and (often) core memory.

Recently I learned about the PiDP-11 [2] and PiDP-8 [3] projects, the loving work of Oscar Vermeulen, where Oscar recreated the front panels of a PDP-11/70 and a PDP-8, complete with switches and lights, and made them work with a Raspberry Pi running SimH.

You do not *need* these front panels to explore these architectures, but it makes the exploration that much more interesting and fun … to see the lights blink, to read the registers and memory through them, and to actually change values in the memory through the front panel.

The project includes a directory of different operating systems to run on the PiDP-11 (RT-11, RSX-11m, RSTS/E v7, and a multitude of different Unix systems). The PiDP-8 has operating systems and compilers too, and both have an assortment of games available, although you should not be expecting high performance graphics.

But wait! If you want to really get "down and dirty," there is the Gigatron [4]. The brainchild of Marcel van Kervinck and Walter Belgers, it is a complete CPU and system built out of simple logic gates, the types used before the days of single-chip microprocessors.

This allows you to see how the machine language ones and zeros actually control the CPU, and how few transistors (also known as "gates") can be used to make a CPU.

This is not a machine that can only be programmed in machine/assembly language. You can also run BASIC on it (although you are somewhat limited to the size of the program and variables). There are also many videos, tutorials, and articles on the Gigatron talking about how it was designed and some of the design trade-offs taken.

The Gigatron can also play games, although (again) do not look for 3D graphics – it is estimated at the power of an Intel 8088 computer.

I often tell audiences that people do not buy hardware, and they do not buy software; they buy solutions. People do not glue hardware and boxes of software to their walls with candles on either side like a shrine.

In this case, I will make an exception. I can foresee a small portion of my wall where a PDP-8 and a PDP-11/70, as well as a Gigatron may happily whirl away as a shrine to the early days of computer science, but at a fraction of the electrical power (and cooling) that the original machines used.

Thank you gentlemen, for all your work in recreating these "old" machines. ■■■

Info

[1] SimH: *http://simh.trailing-edge.com/*

[2] PiDP-11:*http://obsolescence.wixsite.com/obsolescence/pidp-11*

[3] PiDP-8: *http://obsolescence.wixsite.com/obsolescence/pidp-8*

[4] Gigatron: *https://gigatron.io/*

Space science for everyone at the Open Source CubeSat Workshop

Tiny Travelers

The CubeSat community gathered in Athens, Greece to explore the possibilities for open hardware and open source software in space. BY RICHARD IBBOTSON

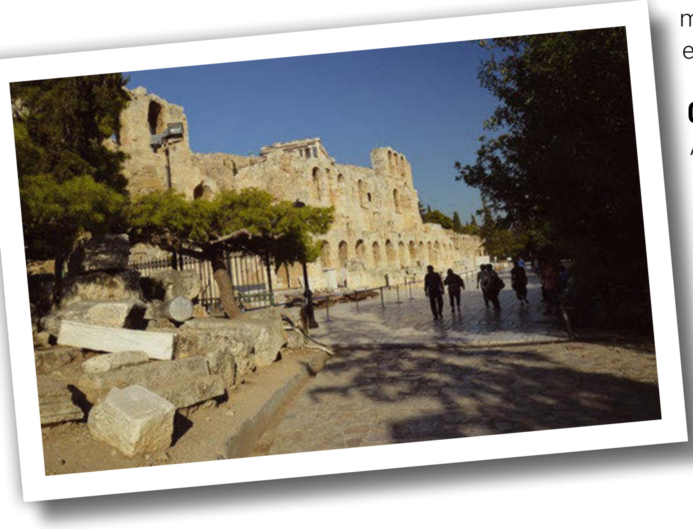
The Open Source CubeSat Workshop took place at the Athens Conservatory in Athens, Greece on October 14-18, 2019 [1]. The LibreSpace Foundation [2], which is sponsored by the European Union and the European Space Agency, organized the event. The general philosophy of the LibreSpace Foundation is to remove the barriers of confidentiality and secrecy involved in satellite development and to facilitate sharing of information on how to build and operate CubeSats.

CubeSats are miniature satellites used for scientific research in space [3] [4]. The most basic CubeSat is a 10cm cube that weighs a maximum of 1.33 kilograms (2.9 pounds). The tiny CubeSat is small enough to ride along as a secondary payload on government and commercial space launches. CubeSats have also been launched from the International Space Station. In all, more than 1,100 CubeSats have been launched into space, carrying a variety of experiments – many created by university students and amateur scientists.

This year's workshop, which had 200 people attend, was focused on developing and applying open source technologies for all aspects of a space mission. The intended audience was academia and research institutes, companies, and individuals. CubeSat enthusiasts who could not make it to the workshop could stream the event from a YouTube page [5].

One of the more interesting presentations was on open source development of low-cost, low-power, sub-joule, micropulsed plasma thrusters, from Michael Bretti, who runs a company called Applied Ion Systems. His talk was well received by an enthusiastic audience. Other topics and presentations at the conference included "Implementation of the CCSDS Service in Python," "Open-Source-Defined Software Radios for CubeSat Operations," and "Interplanetary Mission Analysis."

The SatNOGS State of the Union by Alfredos-Panagiotos Damkalis gave a good up-to-date introduction to what's happening now with the SatNOGS project, which, he explained, is made up of a network of small-scale ground stations that use open source software and hardware. Federico Collado gave a short explanation of a proposal for an open space development program. You can view Federico's presentation online [6].



Conclusion

A good place to learn about CubeSats is the SatNOGS project site [7]. Another interesting source is the European Space Agency's SDR Makerspace site [8], which brings together radio amateurs, GNU Radio developers, and software-defined radio experts to work on building open-source satellite communications solutions.

An easy way to find the talks from this year's workshop is to go to YouTube and search for "Open Source CubeSat Workshop 2019." And, if you decide you are interested, drop by the 2020 CubeSat Developers workshop on May 4-6, 2020 in San Luis Obispo, California. ■■■

Info

[1] Open Source CubeSat Workshop 2019: *https://oscw.space/*

[2] LibreSpace Foundation: *https://libre.space/*

[3] Wikipedia on CubeSat: *https://en.wikipedia.org/wiki/CubeSat*

[4] NASA CubeSat Launch Initiative: *https://www.nasa.gov/directorates/heo/home/CubeSats_initiative*

[5] Open Source CubeSat Workshop 2019: *https://indico.oscw.space/event/3/*

[6] Space Development Program "Edu CubeSat" by Federico Collado: *https://youtu.be/NFNM0PHtwEI*

[7] SatNOGS Project: *https://satnogs.org/*

[8] SDR Makerspace: *https://sdrmaker.space/*

Kube email
Making Connections

Kube aims to unite your email, contact lists, calendar, and other online communications in one sleek modern package. BY MICHAEL MCCALLISTER

Pundits have been saying and writing that "email is dying" for decades now. I'll bet your inbox provides plenty of evidence that this is not true.

For a while, email's replacement looked like a "personal information manager," with messaging, contacts management, a calendar, task manager, and even an RSS-based newsreader. The idea was to have these applications under one roof (e.g., KDE Kontact, Mozilla Thunderbird, and Gnome Evolution), occasionally interacting with each other, but mostly simplifying these related tasks for the human who runs them.

The Kube project [1] is building a Qt-based "modern communication and collaboration client," with another organizing principle: Blur the lines between all these packages. If you want to interact with Bill Jones, open his profile and decide what you need to do next: Send an email, start a chat, or schedule a meeting. When Kube comes into its own, you'll see the traditional KDE emphasis on building software that works the way you want, with nearly endless customizability.

Kube also has a clean look and feel, showing the team's work with usability specialists to create a sensible workflow. In this article, I'll show you what Kube is looking to create, and what you can do with it today.

The Vision

In the current version of Kube, you can begin to see what lead developer Christian Mollekopf has in mind as Kube progresses as an application [2-4]. With every account you create, you're really creating a view into these various types of task- or communication-related bits of information.

Since Kube initially starts with creating an email account, you'll see the Conversations view, where you can send emails and organize them by thread. Mollekopf would like to see you be able to add chats, blog posts and responses, and other online conversations here. Down the road, think of other ways you communicate with friends, co-workers, customers, and businesses. Kube aims to help you out.

The People view lets you see the people with whom you communicate. Perhaps you sent a request for help from someone recently, but don't think they responded. Open the People view and locate the person. You'll find all your recent meetings and email with that person. If you need to follow up, just click the email message, and you're on your way. The People view will also track your calendar to see what meetings different people were invited to (and perhaps even if they attended!)

Finally, a single Composer view is for writing email, manually adding people, and editing calendar items. In the long view, Mollekopf sees the Composer as simply a content editor, where you could save anything you write here as a note, a social media post, meeting minutes, or a blog entry.

Kube's roadmap highlights more views on the horizon:

- An Organize view with a tree view for all types of content and folders for mail
- A Read view for all types of content (mail, chats, calendar, and to-do lists) with "triage" tools to help you process mail and other content faster and more effectively
- A Do view with the things you need to focus on today
- A Plan view that allows you to create to-do lists, events, invitations, and more
- An Observe view that lets you check a status board for your projects

Some History

KDE has always had a personal information manager included in its default software collection. If your distribution doesn't include it in the default KDE desktop install, you can install the Kontact [5] suite of applications, or any of its separate components: KMail, KAddressBook, KOrganizer (calendar), Akregator (RSS newsreader), and KNotes. These are all stable applications and well maintained.

Kube developers say that their product won't replace Kontact in the panoply of KDE apps, but that it will give users a more modern alternative.

While the development team is mostly composed of Kolab developers, Kube won't replace existing Kolab installations. The Kolab project began in 2002 as a way to expand the personal information suite into the corporate environment, with a client-server setup similar to Microsoft's Exchange Server and Outlook client. They also produce the commercial Kolab Now cloud email service [6].

While the Kube project began in 2015, it didn't really attract attention until 2017, with its 0.3.1 release. A bit rough around the edges, it worked as a basic email client and address book. Mollekopf wrote at the time: "While Kube 0.3.1 still isn't anywhere near production ready, I can finally say that I can use it for most of my email needs."

Version 0.7 came out in July 2018, with threaded Conversations available and the introduction of OpenPGP message encryption. A macOS client joined the roster. One month later, the

Kube project made nightly builds available through a Flatpak installation [7]. Since then, users have seen the ability to manually add people to their address books and add recurring events to the calendar. Version 0.8 is now available, but Mollekopf admits the Flatpak system "makes (releases) significantly less interesting."

Installing Kube

With a few exceptions for distributions with Kube packages (namely Arch, Gentoo, and Fedora Copr), you can download the aforementioned Flatpak package to install the nightly Kube build [8]. Make sure your distro supports Flatpak and then run the suggested commands from the shell to get started:

```
flatpak -y --user install ⮐
  --from https://files.kube-project.com/⮐
  flatpak/com.kubeproject.kube.flatpakref
flatpak run com.kubeproject.kube
```

Kube will integrate into your KDE system menu automatically. The Flatpak gives you the current build whenever you run Kube, though I wish the interface included an *About* link with a build number to track progress. Checking for a new build also tremendously slows down loading Kube.

Setting Up Accounts

On first launch, you can select an email account with which to sync. Kube supports Kolab Now [6], Fastmail [9] (a secure cloud mail system), and Gmail out of the box. See Table 1 for what you'll need to set up Kube.

You will need an app password, which is different from your account password, from Gmail and Fastmail. Theoretically, these app passwords are one-time-use credentials, but Kube doesn't always remember them, so be prepared to create them multiple times. If you have email from another provider (such as your Internet service provider or Yahoo Mail) click *Other* from the panel. You'll need to have the addresses of the incoming (IMAP-only) and outgoing (SMTP) servers to add them.

Kube can also connect to CalDAV calendar servers and CardDAV address book servers through this screen. For example, to connect Kube to your primary Google Calendar, in the CalDAV address field, type

```
https://apidata.googleusercontent.com/⮐
  caldav/v2/<yourusername>@gmail.com/events.
```

Working with Views

Once your accounts are set up, you can send an email. This is the most-developed part of Kube.

Table 1: Setting Up Kube for Your Mail Account

Mail Accounts	What You Need
Gmail	Application password (from Google)
Fastmail	Application password (from Fastmail)
Kolab Now	Kolab Now credentials
Other	Addresses for incoming (IMAP-only) and outgoing (SMTP) servers

Click *New Email*. The Composer window opens, and you type your message in the left pane (Figure 1). There is nothing remarkable here, except that you add your recipient(s) on the right pane, not on top of the message.

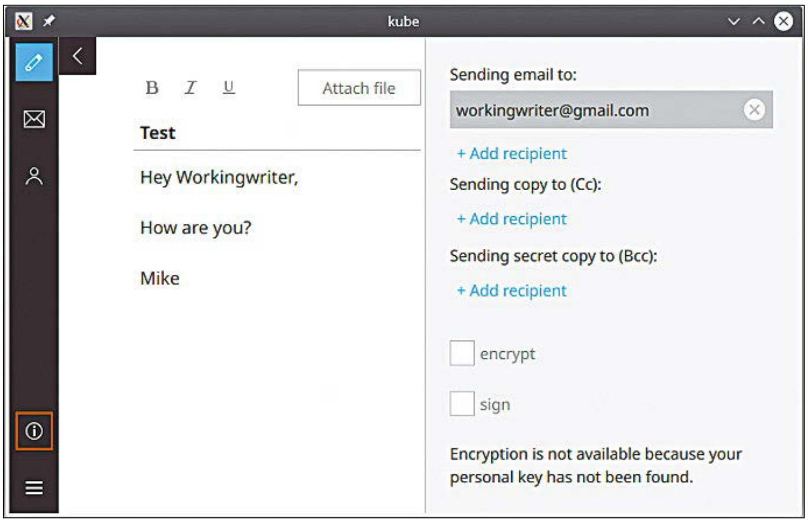
Encrypting Messages

If you're using a cloud-based email service (and who isn't these days), you may want to increase your use of encrypted messaging. Some messages you send may contain more sensitive material, but that's not the only reason to secure your communications. You may want to keep the forces of surveillance capitalism away from your data to protect yourself from your mail service's ravenous appetite for ways to predict your future behavior. Kube offers a basic solution to this problem.

If you have need for secure communications, Kube supports end-to-end GPG encryption and signing for all email messages. The support is rudimentary at this stage, but expect improvements soon. When you launch the Composer window to send email, Kube will check the local PGP key store for the private key associated with the email address you're using. If it doesn't find the private key, the *encrypt* and *sign* checkboxes will be disabled. When enabled, check one or both boxes to secure your message.

Of course, you will want your recipient (and no one else) to be able to read your encrypted mes-

Figure 1: Kube's Composer window may take getting used to, but you'll find all your email metadata on the right side of the message.



Test

Hey Workingwriter,

How are you?

Mike

Sending email to:

workingwriter@gmail.com

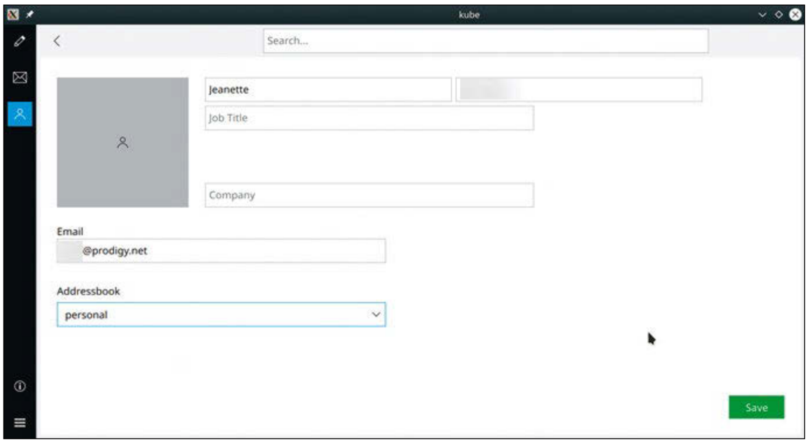
Sending copy to (Cc):

Sending secret copy to (Bcc):

□ encrypt

□ sign

Encryption is not available because your personal key has not been found.



Jeanette

Job Title

Company

Email

@prodigy.net

Addressbook

personal

Figure 2: Use the New Contact page to augment your address book.

sage. Make sure that you have your recipient's public key. Notice the padlock icon to the right of your recipient; if you have their public key in your local store, the padlock will be closed, and you can click the *Send* button. If the padlock is open, you cannot send an encrypted message.

One more caution: Kube does not currently check public keys for validity (i.e., whether it is expired, revoked, or not trusted). Check with your recipient to confirm the key is valid before sending your message.

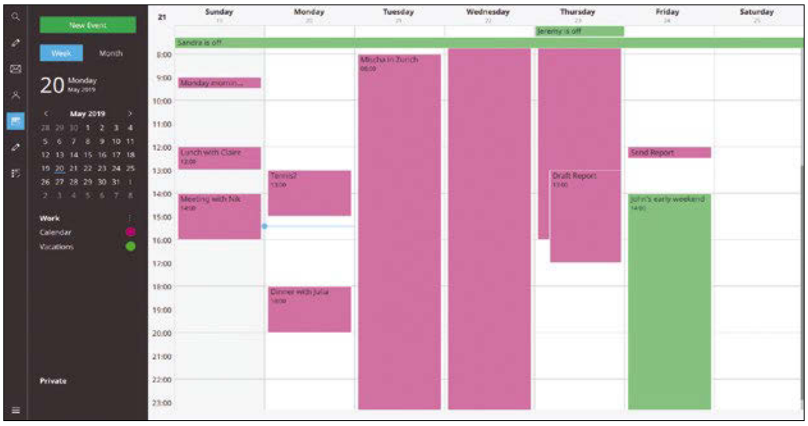
Other Email Options

You can share a received message with others. For now, "share" just means "forward." Down the road, you may be able to click *Share* to place a message on social media, other messaging services, and the like.

Also note that sending a "secret copy" to a recipient in this window is just a blind carbon copy (bcc); the recipient's name and address won't appear in other recipients' mailboxes. You can, of course, send them an encrypted message with their public key.

In the Conversations message list, you can also delete emails, mark them as *Important* (click the star), and change the *Read/Unread* flag (click the envelope). You cannot yet create folders or other organizing tools for your mail, but these triage tools make a good start in helping you to flag priorities.

Figure 3: The *Calendar* view is one of the newest additions to Kube.



Managing People

To add a new contact , click *People* and then *New Contact*. There is nothing odd here (Figure 2): name, photo, company, and job title. Click *Add Email* to add their email address, and you're done. At this time, you can't create multiple address books in the app.

If you can connect to an online address book via CardDAV, this address book should be listed in the People window.

Managing Calendar in Kube

Kube's calendar got some attention in the 0.8.0 release. It will import CalDAV data from other calendar apps (like Google Calendar) and display them (Figure 3). But in the newer version, you can add your own events with a simple entry box. Enter the date, start and end times and a description. You can even invite people in the event!

Summary

Kube is a work in progress with an interesting idea. Most email clients were first developed 20 years ago, and their user interfaces haven't changed much. Kube's design reflects the modern standards it represents. Beyond just being nice to look at (often an overrated quality in revamped software), Kube incorporates important usability advances of recent years. We can all hope for improved functionality and more rapid development. The roadmap is promising. When the "soon" and "later" features become "now," this could be an app I could live in, whether email dies or not! ■■■

Info

[1] Kube: *https://kube-project.com*

[2] Christian Mollekopf's blog: *https://cmollekopf.wordpress.com/*

[3] Kube Documentation: *https://kube.readthedocs.io/en/latest/*

[4] Kube Project on Phabricator: *https://phabricator.kde.org/project/view/43/*

[5] Kontact: *https://kontact.kde.org/*

[6] Kolab Now: *https://kolabnow.com/*

[7] Flatpak and releases: *https://cmollekopf.wordpress.com/2019/05/25/last-months-in-kube/*

[8] Download Kube: *https://kube-project.com/getit.html*

[9] Fastmail: *https://www.fastmail.com/*

The Author

Michael McCallister has been writing email since 1992 and using Linux since the start of the millennium. Among other things, he wrote *openSUSE Linux Unleashed*.

What?!
I can get my issues SOONER?

Available anywhere, anytime!
LINUX MAGAZINE
AUTOMATION TRICKS
LINUXVOICE
WWW.LINUX-MAGAZINE.COM

Organize your photo collection with Geeqie

Art Sorted

The image viewer Geeqie is used to view and sort image collections. The tool supports numerous formats, reads metadata, and – among other things – displays the location where you took the picture on a map.

BY TIM SCHÜRMANN

Smartphones above all, lure holiday makers into taking huge numbers of spontaneous snapshots. At home, you then face the task of filtering out the unusable photos of the church with half a steeple and the unknown cyclist who photobombed your landscape. Geeqie [1] is a lean image viewer that can help you with this. It was created as an enhancement of the GQview [2] image viewer, which was discontinued in 2006, and is now found in the repositories of many distributions.

Unlike many competitors, Geeqie is extremely fast and is easy to use. When viewing the data, a zoom function and a preview with different modes are a big help. The application also starts a slide show at the touch of a button. The "File Formats" box shows which image types the software can handle in version 1.5 or newer. Actions such as deleting, moving, and renaming are carried out via the user interface, usually by drag and drop. Virtual photo albums let you collect topically related images.

Furthermore, Geeqie provides insights into the metadata of the images if they are in EXIF, IPTC, or XMP format. If necessary, you can add tags to the photos and rate them. If the metadata contain the snapshot location's GPS coordinates, the program visualizes them as a point on an OpenStreetMap map.

Geeqie has a deduplication feature that even compares the content of images when needed. In contrast to its competitors, however, the functions for post-processing the images are limited. They are essentially limited to rotating in 90-degree steps, mirroring, and flipping. You require external tools for all further changes.

File Formats

Geeqie 1.5 supports the following file formats [3]: 3FR, ANI, APM, ARW, BMP, CR2, CRW, CUR, DNG, ERF, GIF, ICNS, ICO, JPE/JPEG/JPG, KDC, MEF, MOS, MRW, NEF, ORF, PEF, PTX, PBM/PGM/PNM/PPM, PNG, QIF/QTIF (Quicktime Image Format), RAF, RAW, RW2, SR2, SRF, SVG/SVGZ, TGA/TARGA, TIF/TIFF, WMF, XBM, XPM, HEIF (primary image only), WEBP, DJVU, animated GIF, and the stereoscopic image formats JPS (side-by-side) and MPO.

Included

Most distributions let you install Geeqie via the software manager. If you fail to find the program there, go to the project homepage and make your choice of distribution in the *Download and Installation* section below the *Linux* heading. Armed with this information, the server then sends you to a page with a corresponding package. However, in some cases, it will redirect to an older version or to a nonexistent page.

If there is no package for the distribution you are using, you can pick up the source code from GitHub [3]. How to build and install the program is explained in the "Build Helper" box. In this case, you are responsible for updates during manual installation.

Trinity

After launching, the three-panel main window appears (Figure 1). In the top left corner, navigate to the directory with the photos that Geeqie lists bottom left. As soon as you click on an image, it will appear on the right side. The directory tree in the upper left corner may be a little too small for your liking. To adjust the size of the three panels, mouse over the separator. As soon as it turns into a double arrow, drag and drop the separator.

If you will be working in the same directory regularly in the future, it makes sense to bookmark it. Click on the plus sign and assign a name to the new shortcut at the top, for example, *Cologne 2019*. Navigate to the desired folder in the list and press *OK*. The shortcut appears in the top left corner of the list above the green plus sign; after clicking on it, the software takes you directly to the linked directory.

The list of photos in the lower left corner is sorted alphabetically by file name. To change this, click on *Sort by Name* in the status bar and select a new criterion. The options in the status bar include the memory space the photos occupy in the directory, the dimensions in pixels, and the file size

Listing 1: Installation, Part One

```
$ sudo apt install libgtk-3-dev liblcms2-dev libexiv2-dev liblirc-dev lirc
libchamplain-gtk-0.12-dev libchamplain-0.12-dev libclutter-1.0-dev liblua5.1-0-dev
librsvg2-common libwmf0.2-7-gtk gawk markdown libffmpegthumbnailer-dev libpoppler-glib-dev
libimage-exiftool-perl liblcms2-utils imagemagick exiftran gphoto2 ufraw exiv2 libheif-dev
libwebp-dev libdjvulibre-dev intltool
```

Build Helper

To compile Geeqie from source code, first install the following tools and libraries, including the respective developer packages: GTK+3, LCMS2 2.0, Exiv2 0.11, Lirc, Libchamplain-gtk 0.12, Libchamplain 0.12, Libclutter 1.0, Lua 5.1, Librsvg2-common, Libwmf0.2-7-gtk, Awk, Markdown, Libffmpegthumbnailer 2.1.0, Libpoppler-glib-dev 0.62, Libimage-exiftool-perl, Liblcms2-utils, ImageMagick, Exiftran, gphoto2, ufraw, exiv2, Libheif, Libwebp, Libdjvulibre, and ZoneDetect.

On Ubuntu, the command from Listing 1 imports everything you need. Download the archive with the source code of the current Geeqie version from GitHub [4]. Unpack it on your hard disk, change to the directory created in a terminal window, and call the commands from Listing 2 there. The first command prepares for the build and warns you if a dependency is missing. In this case, use the package manager to install the required software, and then call `./autogen.sh` again. Finally download the time zone data [5] and copy the data to the `/usr/local/lib/geeqie/` directory.

Listing 2: Installation, Part Two

```
$./autogen.sh.
$ make
$ sudo make install
```

of the currently displayed image. In another field Geeqie reveals the zoom level at which it is currently showing the photo.

Matter of Opinion

Geeqie rotates photos independently if the metadata contains alignment information. To disable this automation, select *Edit | Orientation | Exif Rotate*. The other menu items under *Edit | Orientation* let you rotate, flip, and mirror the photo manually.

Geeqie tries to display the colors correctly using the color profile contained in the photo. If this goes wrong, disable the *View | Color Management | Enable Color Management* item in the menu.

The fastest way to control Geeqie is to use keyboard shortcuts. You can also access almost all of the functions via the menu. To zoom in and out of the image, you can select *View | Zoom*, or press *+* and -. *X* fits the image to the window, while *Z* restores it to the original size. To go to the next image in the list, click on the image or press the space bar. In the image stock, you can navigate with the mouse scroll wheel, or simply press Up and Down. All other keyboard shortcuts are listed on the page that appears when you select *Help | Keyboard Shortcuts*.

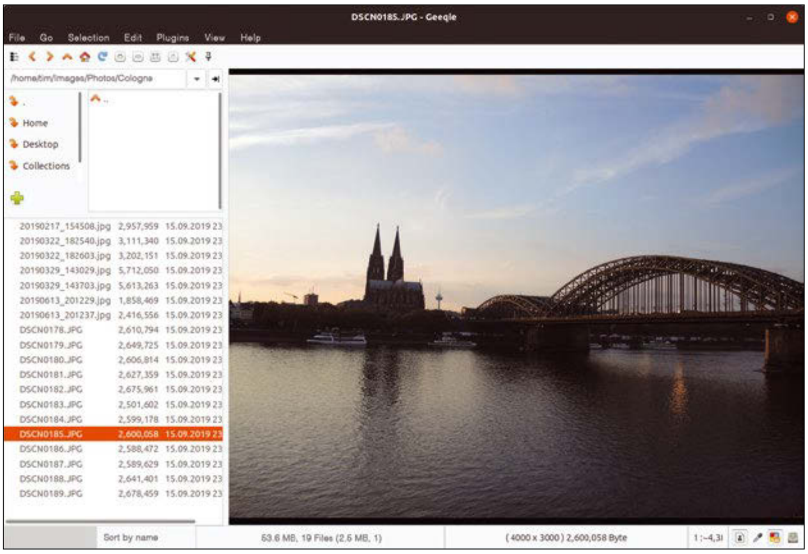
However, this is a time-consuming way of perusing your image stock. You can do this more quickly with the small thumbnails that you can call up by pressing *T*. Ctrl+I shows you a small light table view (Figure 2); Ctrl+L returns you to the list. You can also press *F* to toggle to full-screen mode and back.

Slide Show

In addition, *S* starts and stops a slide show, which runs directly in the main window. Its speed can be adjusted by pressing Ctrl++ and Ctrl+-. If you right-click on a directory, the context menu lets you start a recursive slide show. Geeqie will then include the photos in all subdirectories.

View | Pan View tells Geeqie to display the photos in other overviews (Figure 3), by default as a timeline. *Go to original* in the context menu shifts the corresponding photo into the main window.

Pressing Ctrl+K unfolds a sidebar with more information (Figure 4). Among other things, you can see a histogram of the image, assign keywords,



```
20190217_154508.jpg 2,957,959 15.09.2019 23
20190322_182540.jpg 3,111,340 15.09.2019 23
20190322_182603.jpg 3,202,151 15.09.2019 23
20190329_143029.jpg 5,712,050 15.09.2019 23
20190329_143703.jpg 5,613,263 15.09.2019 23
20190613_201229.jpg 1,858,469 15.09.2019 23
20190613_201237.jpg 2,416,556 15.09.2019 23
DSCN0178.JPG        2,610,794 15.09.2019 23
DSCN0179.JPG        2,649,725 15.09.2019 23
DSCN0180.JPG        2,606,814 15.09.2019 23
DSCN0181.JPG        2,627,359 15.09.2019 23
DSCN0182.JPG        2,675,961 15.09.2019 23
DSCN0183.JPG        2,501,602 15.09.2019 23
DSCN0184.JPG        2,599,178 15.09.2019 23
DSCN0185.JPG        2,600,058 15.09.2019 23
DSCN0186.JPG        2,588,472 15.09.2019 23
DSCN0187.JPG        2,589,629 15.09.2019 23
DSCN0188.JPG        2,641,401 15.09.2019 23
DSCN0189.JPG        2,678,459 15.09.2019 23

53.6 MB, 19 Files (2.5 MB, 1)    ( 4000 x 3000 ) 2,600,058 Byte    1 :~4,31
```

Figure 1: If you launch Geeqie from a terminal window, the program automatically displays the photos from the current directory.

and enter a comment if required. You can either check the boxes for the keywords in the list or enter your own in the field to the left. *Rating* lets you assign a rating to the image.

Figure 2: You can adjust the size of the thumbnails in the *Edit* | *Preferences* | *General* tab.

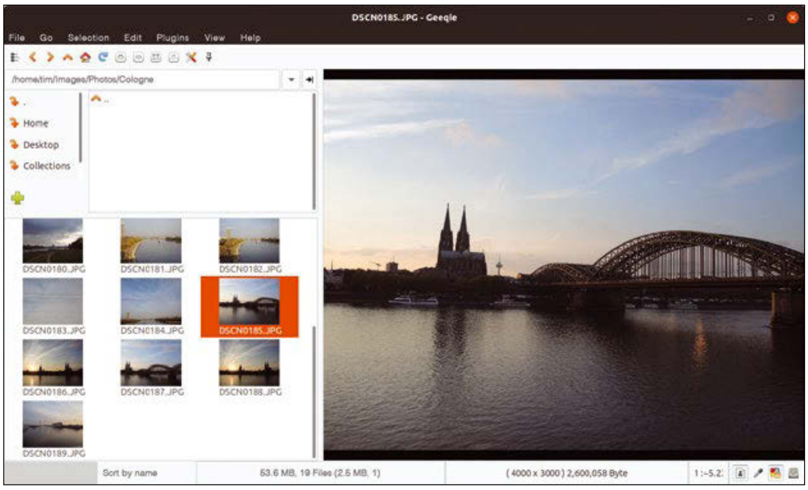


Figure 3: The *Timeline* drop-down list in the panel view lets you switch to this *Calendar*. Clicking on a date shows you all the photos taken on that day.

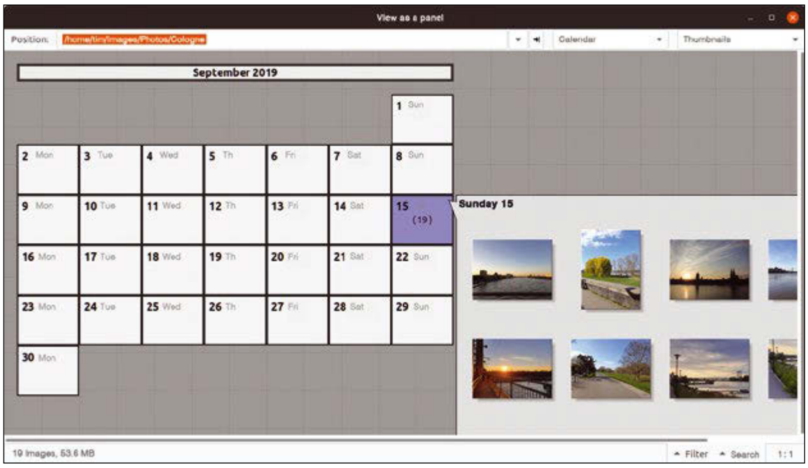
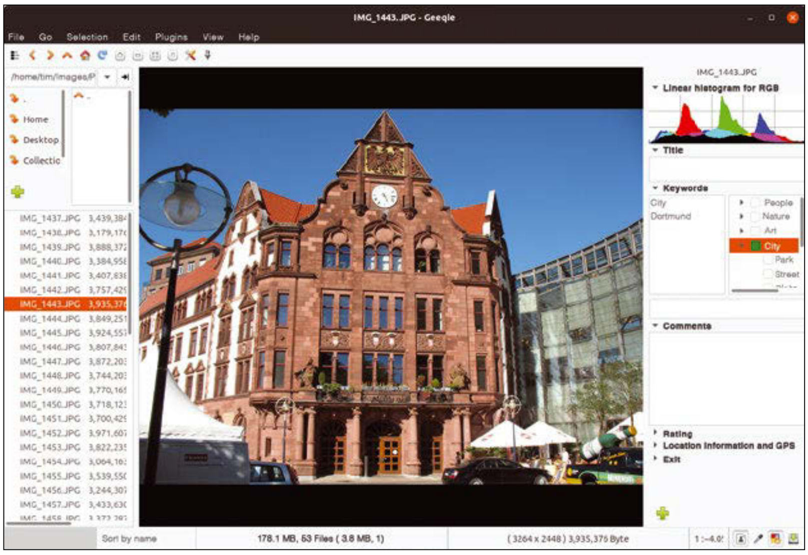


Figure 4: The individual information can be displayed and hidden by clicking on the black triangles.



In the lower part of the information, you will find all the EXIF metadata stored by the camera, as well as the recording location on an OpenStreetMap map. You can add missing information by pressing the plus button. If you cannot access some information, this probably means that Geeqie was compiled without the required support.

Love of Order

The list of photos on the left side and the directory selection work like a file manager: You can drag a photo and drop it in another directory. The application will then ask if you want to copy or move the image. If you hold down *Ctrl*, you can select several photos and move them at the same time. New directories are created with *File* | *New directory*.

If your camera has dumped all the photos in a single directory, the Sort Manager helps you to sort them. To bring it up, press Shift+S. Make sure that the drop-down list shows *Folders*. The Sort Manager offers the option of copying or moving the photos to the desired directories (Figure 5).

You can use the plus symbol at the bottom of the Sort Manager to create a shortcut for each target directory. Select the first photo in the file list in the lower left corner of the main window. Then, in Sort Manager, click the shortcut to which you want to assign the image. Geeqie immediately switches to the next image. Now click on the appropriate shortcut in the Sort Manager again. Continue with this until all the photos are in the desired directory.

In addition, the software offers to assign the photos to virtual photo albums, also known as Collections. In this way, you can keep all the photos of your new house build together, even if the photos actually reside in different directories. To do this, first create a *New Collection* in the File menu; then drag the desired photos into the empty window. Pressing Ctrl+S saves the collection; you can load it later on by pressing *O*. A double-click on a photo opens it in the main window; you can use drag and drop to arrange images in the desired order.

Double Deal

F3 displays the search function on the screen. At the top, first define the directory where you want Geeqie to search. If you select *Recurse*, the tool also looks in subdirectories. Below you can select all the criteria that apply to the photos you are looking for (Figure 6).

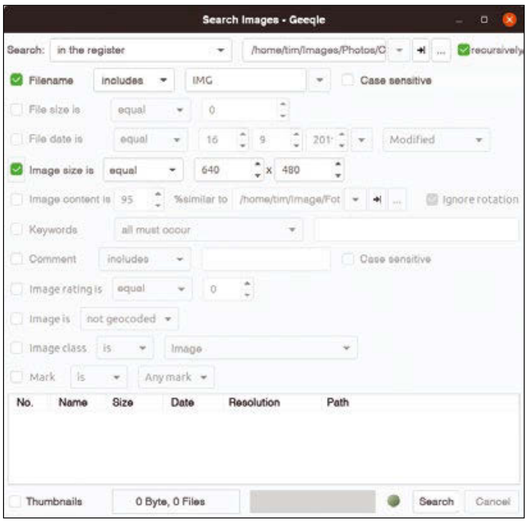
To get rid of duplicate images, call *File* | *Find Duplicates*, and use *Compare by* to select a criterion. *Name* tells Geeqie to report files with identical filenames, while *Similarity* tells the program to report photos with similar image content. Drag all the images you want to compare onto the large white area. The list contains all the photos that Geeqie classifies as similar (Figure 7).

Which sunset snapshot is the best? You answer such questions via *View | Split | Quad*. Geeqie now displays four photos side by side on the right (Figure 8). You can, for example, click on the photo top right and select a different photo from the list bottom left. Changing the two photos at the bottom follows the same principle – this makes it really easy to compare four shots. The program always applies all actions to the photo with the red frame. Pressing *Y* returns you to the normal view.

Pressing *1* tells Geeqie to attach (or remove) a yellow sticky note numbered 1 to the currently displayed photo. You can use the number keys from *2* to *0* to attach up to nine more sticky notes to the photo. What you use the tags for is up to you. For example, you could tag all potential deletion candidates with *1* first.

M shows boxes that represent the individual tags in the bottom left corner of the file list. Checking the first box from the left, assigns the *1* tag to the photo. If you check a box in the top line, Geeqie will only display photos with the appropriate tag. Pressing Ctrl and one of the numbers lets you select all tagged images at once.

Figure 6: In this example, Geeqie would find all images that include the abbreviation *IMG* in their filenames and that measure 640x480 pixels.



Search Images - Geeqie

Search: in the register | /home/tim/Images/Photos/C | ☑ recursively

☑ Filename | includes | IMG | □ Case sensitive

□ File size is | equal | 0

□ File date is | equal | 16 | 9 | 201 | Modified

☑ Image size is | equal | 640 x 480

□ Image content is | 95 | %similar to | /home/tim/Image/Fot | □ Ignore rotation

□ Keywords | all must occur

□ Comment | includes | □ Case sensitive

□ Image rating is | equal | 0

□ Image is | not geocoded

□ Image class | is | Image

□ Mark | is | Any mark

No.	Name	Size	Date	Resolution	Path

□ Thumbnails | 0 Byte, 0 Files | Search | Cancel

Info

[1] Geeqie: *http://www.geeqie.org*

[2] GQview: *http://gqview.sourceforge.net*

[3] GitHub repository for Geeqie: *https://github.com/BestImageViewer/geeqie*

[4] Geeqie source code: *https://github.com/BestImageViewer/geeqie/releases*

[5] Time zone data: *http://files.bertold.org/zd/db.zip*

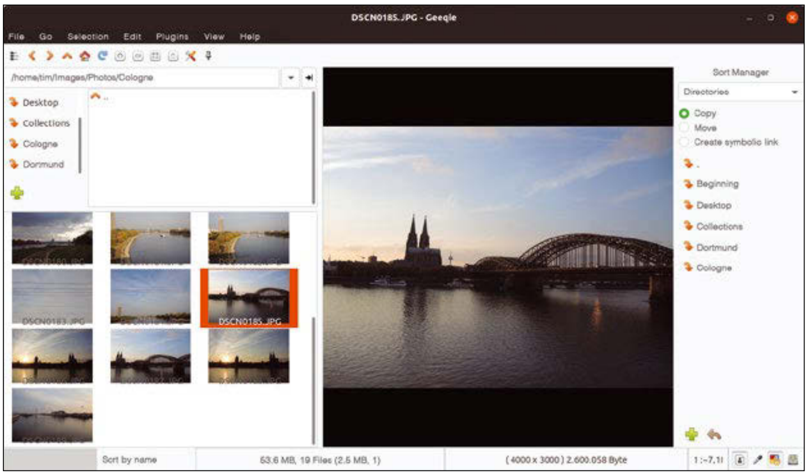
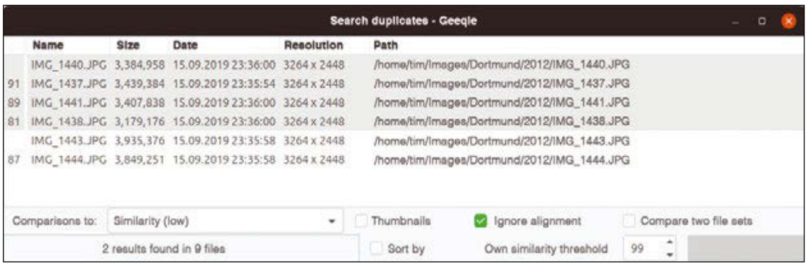


Figure 5: The Sort Manager helps you move or copy the images to directories of your choice.

Conclusions

Geeqie is an extremely fast and stable image viewer with which you can quickly view and sort even large amounts of images. The program works with folders, which you can also access with other image processing programs. You also need to do this if you want to edit your snapshots at some point. Further information on Geeqie's operating principle is provided by a fairly sparse reference accessible via *Help | Contents*. ■■■

Figure 7: In the first column, Geeqie reveals how similar the photos are. The bottom image matches the one directly above it (without a percentage) with a probability of 87 percent.



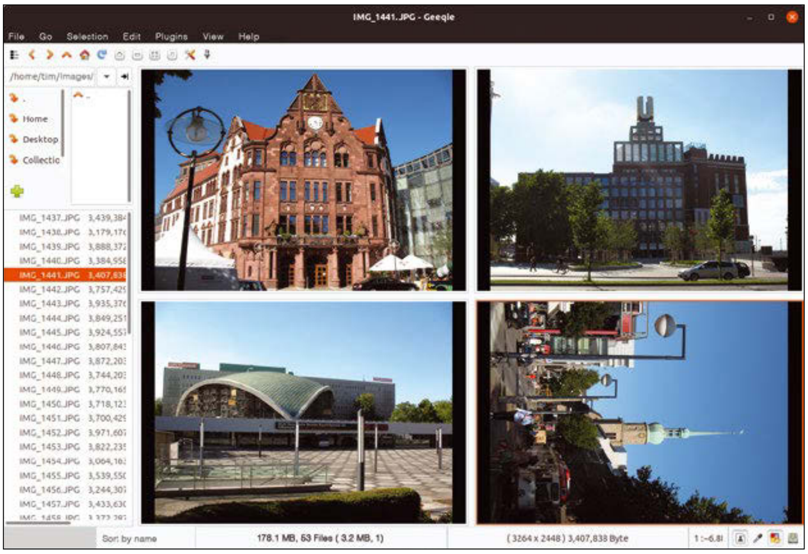
Search duplicates - Geeqie

	Name	Size	Date	Resolution	Path
	IMG_1440.JPG	3,384,958	15.09.2019 23:36:00	3264 x 2448	/home/tim/Images/Dortmund/2012/IMG_1440.JPG
91	IMG_1437.JPG	3,439,384	15.09.2019 23:35:54	3264 x 2448	/home/tim/Images/Dortmund/2012/IMG_1437.JPG
89	IMG_1441.JPG	3,407,838	15.09.2019 23:36:00	3264 x 2448	/home/tim/Images/Dortmund/2012/IMG_1441.JPG
81	IMG_1438.JPG	3,179,176	15.09.2019 23:36:00	3264 x 2448	/home/tim/Images/Dortmund/2012/IMG_1438.JPG
	IMG_1443.JPG	3,935,376	15.09.2019 23:35:58	3264 x 2448	/home/tim/Images/Dortmund/2012/IMG_1443.JPG
87	IMG_1444.JPG	3,849,251	15.09.2019 23:35:58	3264 x 2448	/home/tim/Images/Dortmund/2012/IMG_1444.JPG

Comparisons to: Similarity (low) | □ Thumbnails | ☑ Ignore alignment | □ Compare two file sets

2 results found in 9 files | □ Sort by | Own similarity threshold 99

Figure 8: As an alternative view, Geeqie places two photos one on top of the other or next to each when you press *E* or *U*.


IMG_1441.JPG - Geeqie
File Go Selection Edit Plugins View Help
/home/tim/Images/
Home
Desktop
Collectio
Sort by name
178.1 MB, 63 Files (3.2 MB, 1)
(3264 x 2448) 3,407,838 Byte

FOSSPicks

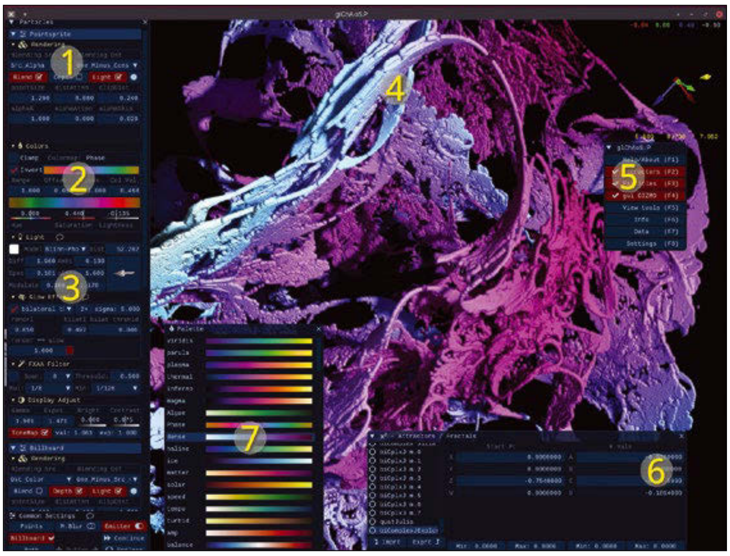
Sparkling gems and new releases from the world of Free and Open Source Software

Graham originally played Sonic the Hedgehog in the early '90s by buying a Sega Genesis/Mega Drive on a rainy Saturday, playing the entire game on Sunday, and returning the console for a refund on Monday. BY GRAHAM MORRISON

Fractal explorer

glChAoS.P

The infinite variety found in fractal geometry is a component in all kinds of practical things, from the generative realistic geometry of islands, rivers, and flora in video games and CGI, to being a principal character in the chaos theory of dynamical systems. But it wasn't so long ago that the beauty of fractals was enjoyed simply for its own sake. You'd launch a graphical fractal renderer on your PC and click around to explore the infinite zoom of a Mandelbrot branch, often cycling through as many garish colors as your graphics hardware could handle. These tools didn't even operate in real time, because the deeper into a fractal you traveled, the longer each beautiful view would take to render, eventually taking hours. One of my favorites was called ChaosPro, running on an Amiga, and some renders would take hours to generate a single frame. But it never detracted from my enthusiasm for these tools.



1. Rendering options: Fractals are rendered as particles, which can be processed in many ways. 2. Dynamic UI: Each palette can be expanded and contracted, with many options for every category. 3. Light: Illumination direction, quality, and shadows have a huge effect on the rendering time and quality. 4. OpenGL output: Your hardware is used to generate the 3D preview. 5. Main menu: Enable and disable parts of the GUI. This works well with the full-screen view. 6. Attractors/Fractals: Open and save example files, switch through presets, and adjust their main parameters. 7. Palette: Just like old fractal viewers, changing the color palette can change everything about the image.

Now that computers and their GPUs are magnitudes faster than those old machines, there doesn't seem to be the same number of fractal exploration applications. However, this is one. glChAoS.P doesn't just share some of its (unpronounceable) name with ChaosPro, it's also an amazing graphical application that almost single handedly makes up for the lack of other options. Rather than letting you explore those old two-dimensional Mandelbrots, glChAoS.P harnesses the power of your GPU to let you explore "strange attractors" and "hypercomplex fractals." A strange attractor, like that chaos theory of old, is a structure that happens to exhibit a complex structure from a more easily defined set of initial conditions. glChAoS.P includes over 60 different kinds, and they all look considerably different. A hypercomplex fractal, new to recent releases of glChAoS.P, usually consists of three (or more!) dimensions. The mathematics behind all of these looks horrendously complex, but you don't need to understand it. This is because glChAoS.P does such a great job of making you want to experiment and play with the various parameters it offers while it continuously renders a perfectly realistic viewport on this unimaginably complex realm.

glChAoS.P is an acronym for OpenGL Chaotic Attractors of Slight (dot) Particles, and this helps to explain a little how the amazing graphics are rendered. The application can handle 100 million particles in less than 1.6GB of RAM, and these particles are used to visualize the structures, alongside options for glow effects, colors, 3D blending/solid, or lit particles. All of which combine to create super-realistic projections of totally alien sculptures, rendered in real time and completely manipulatable by the various sliders to control the fractals themselves. You can easily experiment with a view and navigate just as you would with any other application. It's also easy to navigate between the various demo files, and all of the parameters have their own sliders, color previews and values that can be dragged to modify them. The results are amazing, and if you've got the graphics hardware to drive it, you've got to take it for a spin!

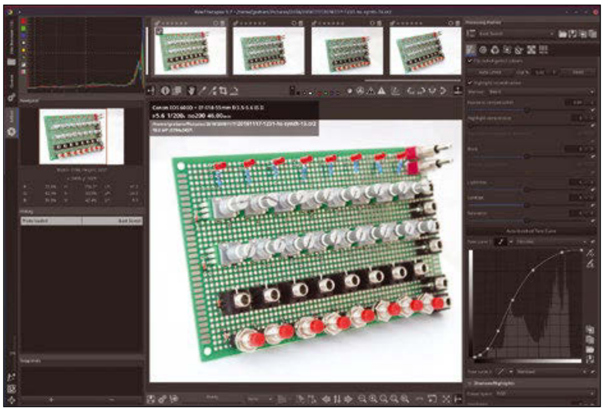
Project Website
https://github.com/BrutPitt/glChAoS.P

Image processing

RawTherapee

When it comes to raw digital photo processing, RawTherapee has always been part of the open source revolution and helps many professional photographers sidestep Adobe's hegemony. But RawTherapee doesn't often get the same attention as darktable, which is a pity, because the two applications together are responsible for revolutionizing photo processing on Linux. If you're looking for a quick comparison, RawTherapee is slightly more Adobe-like, letting you do more within a single application than darktable's focus on post processing. In RawTherapee, it's easier to browse your photo collection via its file manager, for example, rather than scanning them into an internal database, and it will load hundreds of thumbnail previews from a randomly chosen folder much quicker. This makes it great for the unorganized photographer or perhaps the ambitious amateur, who hasn't sorted out their sorting yet.

RawTherapee also feels quicker when editing. It contains many of the same functions as darktable, such as exposure, shadows, darkness, sharpening, color correction, chromatic aberration removal, lens compensation, and many more, but their parameters are immediately accessible and tweakable, unlike the preset-enable-plugin nature of darktable. The effects aren't perhaps as subtle and are not as good at correcting photographic problems, such as underexposure or high contrast. But with careful editing of good source material, the output is almost as good as darktable's while using fewer PC resources. This is important if all you want to do is edit a virtual reel of holiday photos, rather than produce images for print or syndication. RawTherapee is also an application that's developing at an incredible rate, with version 5.7 adding support for film negative processing, image rating, and dozens of bug fixes. This means it's not so much about choosing one application, but simply having the choice to select the best open source tool for the job at hand.



RawTherapee might not have the same fine-tuning as darktable, but it's faster to preview and navigate hundreds of images.

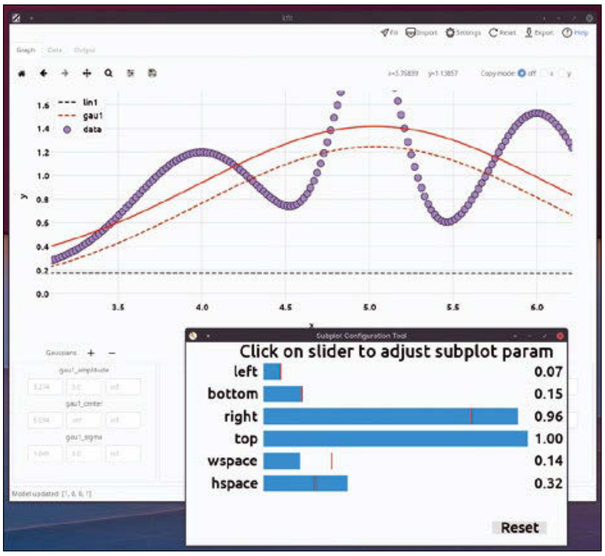
Project Website
https://rawtherapee.com

Spectral fitter

kfit

Spectral diagrams typically show changes in amplitude across the frequency range of a sample or measurement. In audio, for example, they can plot the frequency level of a slice of sound across the human audible spectrum. It's a great way to visualize a spike at 60Hz due to hum from the power supply or the 10KHz whine of a faulty capacitor. But in more scientific analysis, a spectral diagram might show just a single frequency with a few harmonics, and it can then be useful to create a function to describe the observed values outside of the plot. This is what the spectral fitter kfit does. You start by adding the data readings for your spectral plot. There's a tab for this, with columns for *X* and *Y* values, but you can also import a CSV file of external measurements. The plot is then drawn in the main window and you can start to create your fit.

Kfit has four fitting models: Gaussian, Lorentzian, Pseudo-Voigt, and linear. They're added using the plus and minus buttons for each type beneath the graph. You can add more than one at a time to get your fit as close as possible, and they each require a different set of parameters as you try to create something that fits. After adding the values you wish to try, you click on the *Fit* icon in the toolbar. This will draw your new curves across the original spectrogram. You can also change the size and scale of the workspace using a floating panel. When you do have output you can work with, it can also be exported via a CSV, or the data can be viewed from the main window's *Output* tab. That's about all there is to it – it's not as complex as using something like Mathematica, but its limitations make it ideal for use in education or if you just want to play around.



Click on slider to adjust subplot param
left 0.07
bottom 0.15
right 0.96
top 1.00
wspace 0.14
hspace 0.32

Alongside being able to import and export data, you can save the graphs themselves as a bitmap or vector image.

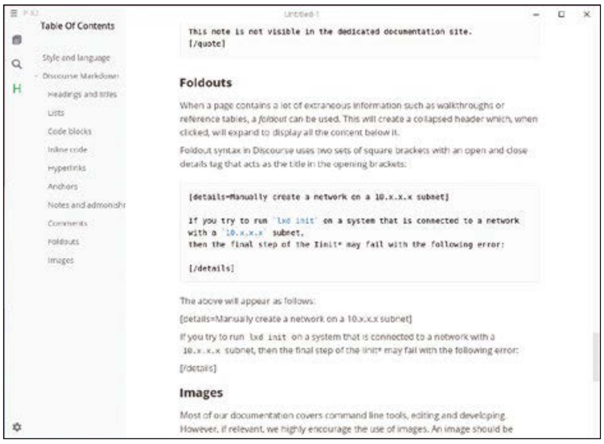
Project Website
https://github.com/rhroberts/kfit

Text editor

Mark Text

That there are so many text editors designed to work exclusively with Markdown says a lot about the continued and still growing popularity of Markdown. It also says a lot about Markdown's disparate usage and syntax when there are so many ways of editing and publishing with it. Mark Text, however, is one of the best Markdown editors we've used, and it makes up for Markdown's shortcomings by playing to its advantages – it's a quick, minimal, and expressive way of writing documentation, from blog posts to complete reference guides.

The best thing about Mark Text is that, by default, it just gets out of your way. It opens with a blank page, and you can start writing immediately. Even without engaging the *Distraction Free* mode, which highlights only the paragraph you're working on, there's very little window decoration. All of Mark Text's considerable configurability is hidden behind the hamburger menu in the top-left corner. You don't even need to use this if you don't want to. As you write Markdown syntax words, the editor silently replaces them with the final output rendered equivalent. Headings, lists, comments, and blocks of code are beautifully colored and can be easily changed by selecting one of the equally attractive light or dark themes. Select a paragraph you've written, and a discrete symbol can be used to transform your words into a heading, a math formula, or a code block. Word, character, and paragraph totals update in real time and are shown next to the hamburger menu, which is all the information most writers need. But you can also open the side panel to show the outline and even enable the *Source Code* mode to disable the automatic output rendering. It's brilliant and beautiful, and it makes you want to write.



```
This note is not visible in the dedicated documentation site.
[/quote]
```

Foldouts

When a page contains a lot of extraneous information such as walkthroughs or reference tables, a *foldout* can be used. This will create a collapsed header which, when clicked, will expand to display all the content below it.

Foldout syntax in Discourse uses two sets of square brackets with an open and close details tag that acts as the title in the opening brackets:

```
[details=Manually create a network on a 10.x.x.x subnet]

If you try to run `lxd init` on a system that is connected to a network
with a `10.x.x.x` subnet,
then the final step of the Iinit* may fail with the following error:

[/details]
```

The above will appear as follows:

[details=Manually create a network on a 10.x.x.x subnet]

If you try to run `lxd init` on a system that is connected to a network with a `10.x.x.x` subnet, then the final step of the Iinit* may fail with the following error:

[/details]

Images

Most of our documentation covers command line tools, editing and developing However, if relevant, we highly encourage the use of images. An image should be

Despite its apparent simplicity, Mark Text contains advanced features like image upload, quote completion, and text direction modification.

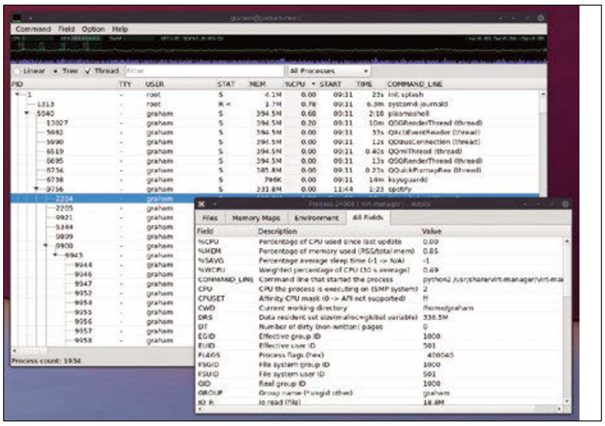
Project Website
https://marktext.app/

Task monitor

qps

Process managers have been around for as long as computers could run more than one process. We don't often think about them. The humble `ps` command is often hidden away behind a pipe and a `grep`, and `top` or `htop` are used when you need something more interactive. On the desktop, all the major environments include their own process managers, so there seems very little reason to use anything else. But what if one desktop's tool does a better job than others? That's exactly what happened with LXQt's qps, which can be built separately without needing any other parts of the LXQt desktop. It's great to see that developers are still creating alternatives and new ways of looking at the same data, which is exactly what qps, a Qt process manager, does, albeit with a Qt-based GUI.

On the surface, qps looks a lot like the output from the `top` command, encapsulated within a GUI. CPU, memory, and swap usage meters are shown at the top with the list of running processes shown below. This list can be formatted as a tree and also show threads; you can click on columns to sort by user, memory usage, CPU, and time. The great thing about doing all this from a GUI is that you don't need to remember the key commands for the function, as everything is just point and click. If you double-click a process another pane appears – this time showing more in-depth details about a particular process. These include the files and sockets being accessed by a process, memory maps and offsets for the libraries being used. Another useful tab in this pane shows a list of all the environment variables accessible to the process, which is a great way of troubleshooting problematic processes or applications, especially when they're being run from user accounts other than your own.



```
PID          TTY  USER    STAT  MEM     %CPU  START  TIME   COMMAND_LINE
▾─1          -    root    S     4.1M    0.00  09:11  23s    init splash
  ─1313      -    root    R <   1.7M    0.78  09:11  6.3m   systemd-journald
 ▾ 5040      -    graham  S     394.5M  0.60  09:11  2:18   plasmashell
   ─13027    -    graham  S     394.5M  0.20  09:11  10m    QSGRenderThread (thread)
   ─5082     -    graham  S     394.5M  0.00  09:11  35s    QXcbEventReader (thread)
   ─5090     -    graham  S     394.5M  0.00  09:11  12s    QDBusConnection (thread)
   ─6519     -    graham  S     394.5M  0.00  09:11  0.40s  QQmlThread (thread)
   ─6695     -    graham  S     394.5M  0.00  09:11  13s    QSGRenderThread (thread)
   ─6734     -    graham  S     385.8M  0.00  09:11  0.23s  QQuickPixmapRee (thread)
   ─6738     -    graham  S     796K    0.00  09:11  14m    ksysguardd
  ▾─9756     -    graham  S     331.8M  0.00  11:44  1:23   spotify
     2204    -    graham
     ─2205   -    graham
     ─9921   -    graham
     ─5344   -    graham
     ─9899   -    graham
    ▾9900    -    graham
     ▾─9943  -    graham
        ─9944  -  graham
        ─9946  -  graham
        ─9947  -  graham
        ─9952  -  graham
        ─9954  -  graham
        ─9955  -  graham
        ─9956  -  graham
        ─9957  -  graham
        ─9958  -  graham
Process count: 1936
```

Process 24908 (virt-manager) - details

Field	Description	Value
%CPU	Percentage of CPU used since last update	0.00
%MEM	Percentage of memory used (RSS/total mem)	0.85
%SAVG	Percentage average sleep time (-1 -> N/A)	-1
%WCPU	Weighted percentage of CPU (30 s average)	0.49
COMMAND_LINE	Command line that started the process	python2 /usr/share/virt-manager/virt-ma
CPU	CPU the process is executing on (SMP system)	2
CPUSET	Affinity CPU mask (0 -> API not supported)	ff
CWD	Current working directory	/home/graham
DRS	Data resident set size(malloc+global variable)	336.5M
DT	Number of dirty (non-written) pages	0
EGID	Effective group ID	1000
EUID	Effective user ID	501
FLAGS	Process flags (hex)	400040
FSGID	File system group ID	1000
FSUID	File system user ID	501
GID	Real group ID	1000
GROUP	Group name (*=egid other)	graham
IO_R	io read (file)	18.8M

Processes can also be listed as threads and trees, which helps when you don't know which has spawned what.

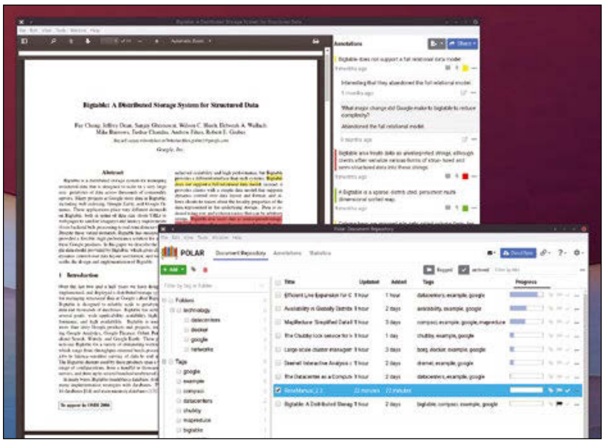
Project Website
https://github.com/lxqt/qps

Text manager

Polar

We probably read more words now than at any other point in history, and it's a pity that we don't collect and curate words in the way libraries do. This is obviously because most of our reading is from the Internet, and consequently, transitory and disposable. But not everything online is like that, and Polar is a tool that can help with reading organization and curation. It's especially helpful when you're conducting any kind of research or study, because it's likely that your data store would become a mishmash of notes, annotations, printouts, and PDFs, with nothing to group them together effectively or track your work. These are all problems that Polar hopes to solve.

Polar is an application designed to help you store online files offline, manage those files, add your own notes and annotations, and sync your entire library with an optional online account. It can deal with PDFs and ePubs, documents can be tags and organized into folders, and Polar lets you add annotations anywhere and export these as Markdown for your own notes. There's a wonderful *Statistics* tab that gives you an overview of everything you've done and when; it feels very much like GitHub's activity tracker for developers. If you're working on a thesis or studying a complex subject, this kind of oversight is invaluable if you need to read through an entire body of work. But even those of us with more modest requirements will find Polar useful. The document reader with its progress bar and integrated annotation is clear and useful even for casual reading. Its focus is always on the value of incremental reading. Having an application that tracks reading like this – rather than a collection like Calibre – is a refreshingly different approach to reading even online documentation.



If you have a large online reading list and love making notes, Polar is a great way to organize a collection.

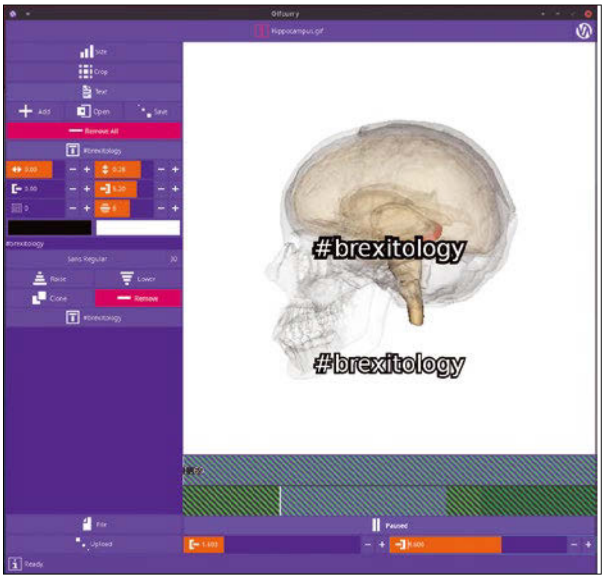
Project Website
https://getpolarized.io/

Meme maker

Gifcurry

There is no escaping the humble GIF. Even in the late '80s and early '90s, there was still no escaping the humble GIF, and it's remarkable that these low-quality meme generators have survived so long. You simply can't be both young and online without inserting one into a chat or social network as a shorthand for some indefinite emotional state or exclamation. Gifcurry is an editor for such things. If FFmpeg is installed, it lets you load up almost any video and turn it into your own low-quality synopsis of a feeling – or a kitten playing on a trampoline. As you'd expect, the interface is as garish as the technology itself and feels like it might even have been designed in the early '90s. But this also makes it easy to use. There's a small toolbar that lets you resize or crop the video, upload it to Giphy or Imgur, and even add a text annotation so that you don't have to rely fully on your audience's sarcasm sensors. The text editor has controls for font size and position, and you can add or duplicate as many as you need.

The actual editing controls are beneath the preview pane. These are simple too. There are controls to let you pause and play back your animation, scroll through its various frames, or step through or back one frame at a time. Editing consists of using scroll bars and +/- buttons to set the beginning frame and the end frame of your animation. That's all there is to it! You can't add transitions, merge two or more animations, or use a clipboard palette to store and rearrange your edits. It's quick, painless, and almost dependency free. It's simple, but that's exactly why GIFs are still around and more popular than ever.

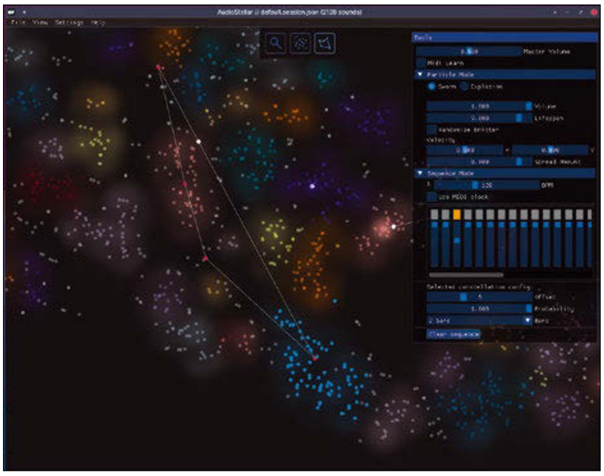


Quickly create your own GIF animations and memes with a graphical tool straight from the '90s.

Project Website
https://github.com/lettier/gifcurry

Sound explorer

AudioStellar

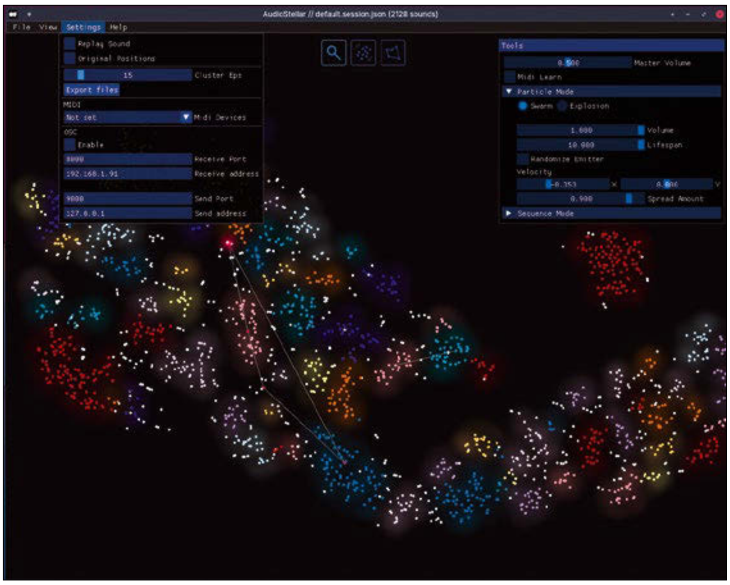


Sequence mode lets you draw lines between different sample points and play them all back together as a huge variable loop.

AudioStellar is a difficult application to categorize. On the one hand, it's a tool that can help you organize your music collection, or at least a huge collection of loops and samples if you're a musician. This is because it uses "machine learning" to listen to your music files and group them into a tree hierarchy you can navigate. On the other hand, it can help you make discoveries – such as finding audio in your collection with a similar frequency grouping and timbre. But it can also operate as a musical instrument in its own right, because you can connect any source of MIDI note data to AudioStellar and play back the sound in your collection. It won't just play back single sounds though, but fire out particles from your mouse cursor, triggering any sound in the tree they happen to hit. In this way, AudioStellar is like a music-exploring, science lab performance instrument from the future.

The set of sounds that are included with the application are a good way to get to know what its capabilities. That's because they're all percussion sounds, and you can easily see where the bass drums are located on the tree, for instance, or the cowbells, or the high hats and cymbals. Click on any point and the audio sample will play back. You can drag and zoom around the audio collection just as you might an online road map. There are three main modes: *Explorer*, *Particle*, and *Sequence*, which are selected by clicking on the magnifying glass, the point cloud, or the constellation icons respectively. *Explorer* is the default mode, and each broad group of sounds is highlighted as you click and move between individual points. *Particle* and *Sequence* mode can both be used to turn AudioStellar into a musical instrument.

Both the aforementioned *Particle* mode, where sounds are triggered by pixels colliding with one another, and the *Sequencer* mode have their own foldable options in a tools palette. This allows you to change the *x* and *y* velocity, the spread, and the volume of the particle, as well as whether they explode from the cursor or swarm. The *Sequencer* options are dominated by 16 sliders and selectors. Select one of these and then draw a polygon by connecting points in the tree, and each point's sample will be triggered in time, playing in a loop. This is a sequence, controlled by the slider for that sequence, and you can have 16 running at the same time. And because they can all be a different length, it's very easy to create polyrhythmic chains that create their own infinitely modulating variations, all synchronized to an external MIDI clock. You can even set an offset and the number of bars for each sequence, potentially changing their timings relative to the other sequences, along with a probability setting that changes the likelihood that any one sound in a sequence is going to be played or not. This creative and performative aspect to AudioStellar is remarkably productive, and a real surprise considering the application legitimately has other more prosaic uses, such as managing and exploring audio similarities. But it all comes together to create an eclectic tool that will hopefully force you to look at even the most staid and conservative music or sample collection in a new way.



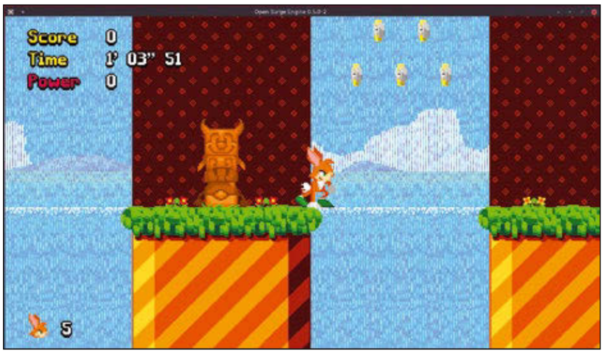
AudioStellar lets you see an audio collection grouped according to similar timbre and even play back sounds from MIDI or OSC.

Project Website

http://audiostellar.xyz/

Modern classic

Open Surge



Open Surge isn't just a paean to a classic platformer, it's also a game designer and scripting engine for creating your own games.

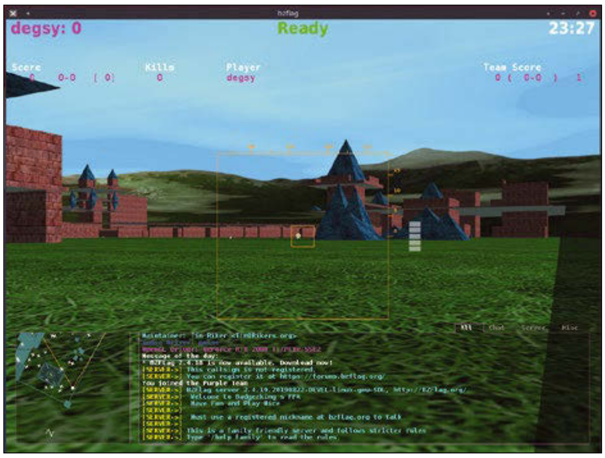
There's no end of renewed-retro consoles being released, finally allowing you to legally play all those old classics. Often, though, the games themselves haven't held up so well to the passage of time. The graphics are seldom as good as you remember, the frame rates abysmal, and the unforgiving permadeath – now you no longer have entire weekends to waste – can feel more sadistic than perfectionist. There's more fun in the nostalgia than the playability. This is why it's great when a new game updates an old mechanic without being encumbered with exactly the same old gameplay. Open Surge does this to the old Sega Genesis/Megadrive classic, Sonic the Hedgehog. It's a game that has many of the same graphical and control elements and uses the same level aesthetic, but is packaged with completely new level and character design.

If you enjoyed the manic adrenaline of Sonic, Open Surge is a great way to revisit a gamestyle that isn't too common. The levels are well designed, the animations brilliantly executed, and the sound incredibly authentic. Its requirements are so low that it will run on a Raspberry Pi, which makes the whole package tantalizingly portable if you include a controller, battery, and screen – three decades after the Game Gear! But perhaps the most important aspect to Open Surge is that it's also a game and level designer. There's a built-in editor that's accessible by pressing F12, and the game's sound, image, character, and scenery assets are easily hackable. It even includes its own scripting engine, SurgeScript, for adding whatever logic you need to each level. It's a brilliant old/new way to get into game design, and one we hope we'll still see around in another 30 years.

Project Website
https://opensurge2d.org/

Multiplayer classic

BZFlag



```
Maintainer: Tim Riker <Tim@Rikers.org>
[illegible]
OpenGL Driver: GeForce [illegible]
Message of the day:
* BZFlag 2.4.18 is now available. Download now!
[SERVER->] This callsign is not registered.
[SERVER->] You can register it at https://forums.bzflag.org/
You joined the Purple Team
[SERVER->] BZFlag server 2.4.19.20190822-DEVEL-linux-gnu-SDL, http://BZFlag.org/
[SERVER->]  Welcome to Badgerking's FFA
[SERVER->]  Have Fun and Play Nice
[SERVER->]
[SERVER->]  Must use a registered nickname at bzflag.org to talk
[SERVER->]
[SERVER->] This is a family friendly server and follows stricter rules
[SERVER->] Type '/help family' to read the rules
```

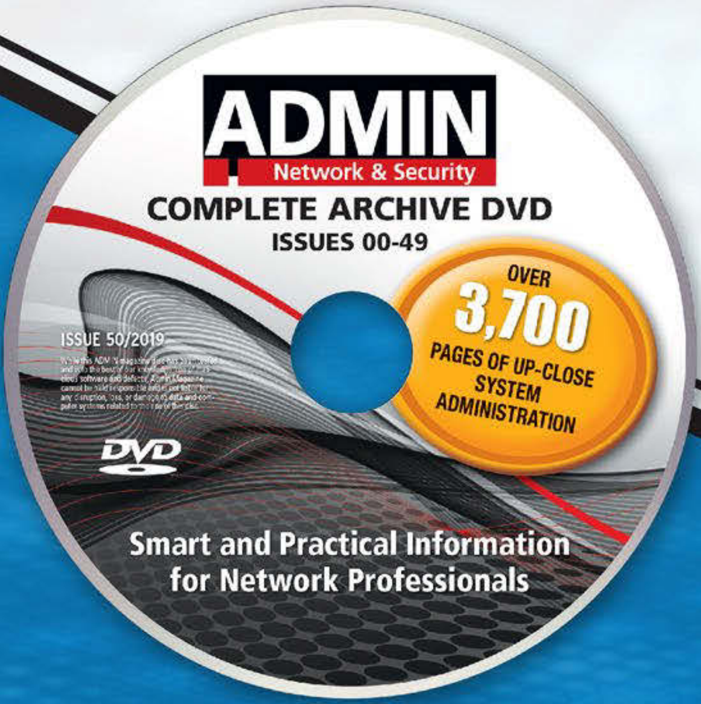
Once a stalwart of every Linux installation, BZFlag can now be installed on Linux, macOS, and Windows, so everyone can join in the fun.

If you first installed Linux in the late 1990s, especially in a university or school lab, one of the first things you might have tried was a game called BZFlag. This was a simple vector game, drawn in lines, and controlled from the player's point of view. You moved within a landscape of simple square buildings that obstructed the distant mountains, where another player would be prowling, waiting to target you in their sights, and blow you into the next level. It was reminiscent of an early vector arcade game called Battlezone, with one big difference: The 1997 release of BZFlag included a multiplayer lobby where you could find games and play on the battlefield with strangers. Despite the retro graphics, it felt incredibly advanced at a time when being online was uncommon.

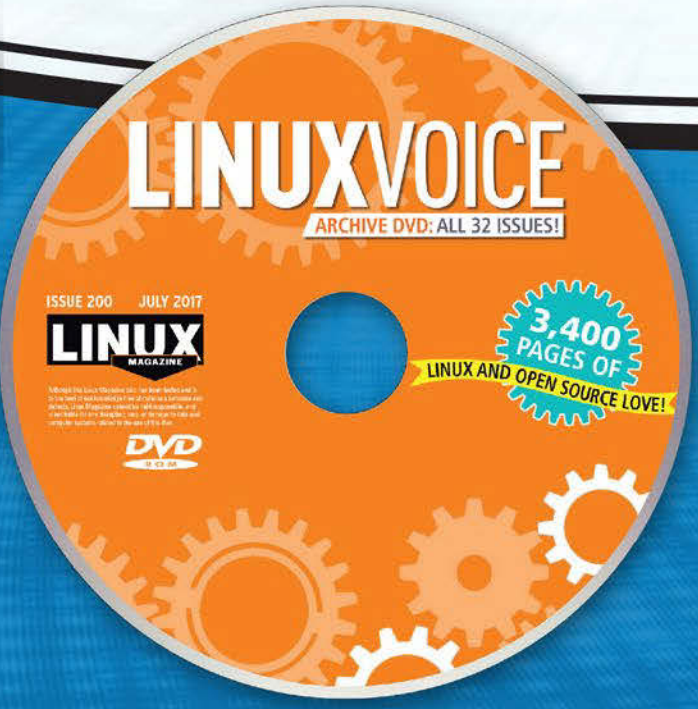
But what many people don't realize is that BZFlag has been in development ever since, with some major updates, a new website, and both Flatpak and Snap options. The graphics, in particular, no longer look like a retro-inspired tribute to a bygone era. There's no realtime RTX raytracing, but there is texture, modern resolution, and lighting support. You can still run the game in both full screen and "quick, minimize, and switch to spreadsheet" window/working mode. Also, the gameplay is just as fun. You can have dozens of players on the same map, and there's usually a server somewhere in the world you can join or run your own server just as simply from the main menu. You'll also find lots of configuration options to fine-tune how the game runs, and it should run on almost any hardware from the last 10 years, even without a dedicated graphics card. But the best thing about BZFlag is what has kept it under development for so long – it's still a huge amount of fun.

Project Website
https://www.bzflag.org/

Searchable Linux Archive: 19,000 Pages of Practical Know-How
18 YEARS OF LINUX! Issues 1-214
All the Tricks
All the Hacks
All the Apps
ISSUE 215
OCT 2018
LINUX MAGAZINE
DVD ROM
THE COMPLETE LINUX MAGAZINE ARCHIVE


ADMIN Network & Security
COMPLETE ARCHIVE DVD
ISSUES 00-49
OVER 3,700 PAGES OF UP-CLOSE SYSTEM ADMINISTRATION
ISSUE 50/2019
DVD
Smart and Practical Information for Network Professionals

THE COMPLETE
UBUNTU
user
ARCHIVE
DVD ROM
Searchable DVD
Content Available in Both HTML and PDF Formats
Over 3,000 PAGES! 7 GREAT YEARS OF UBUNTU USER



LINUXVOICE
ARCHIVE DVD: ALL 32 ISSUES!
ISSUE 200 JULY 2017
LINUX MAGAZINE
DVD ROM
3,400 PAGES OF
LINUX AND OPEN SOURCE LOVE!

Posting text and images to Mastodon

Social Skills

Creating a custom application that toots text to Mastodon (the Fediverse's version of Twitter) is simple and straightforward. But we can mix it up by adding images and video, scheduling posts, and changing privacy settings.

BY PAUL BROWN

If you haven't read the prior article in this series [1], you may want to do that now. There, you will learn how to read from a Mastodon feed and print out the toots (the name Mastodon gives its posts) onto the command line. You can do that quite easily with Python, because the API for Mastodon is open and well documented, and also because there is a comprehensive Python wrapper [2] that makes building ad hoc clients a breeze.

In this article, we're going a step further, and you will see how to create an app that posts your toots. Before you start, and just in case you haven't already, install the `Mastodon.py` module with

```
pip install Mastodon.py
```

and let's get started.

Registering

As with your reading application, the first thing you have to do is register your application with Mastodon (Listing 1).

Listing 1: posttoots_register.py

```
01 #!/usr/bin/env python
02
03 from mastodon import Mastodon
04
05 Mastodon.create_app(
06      'posttoots',
07      api_base_url = 'https://your.mastodon.server',
08      scopes=['read', 'write'],
09      to_file = '.secrets'
10 )
11
12 mastodon = Mastodon (
13     client_id = '.secrets',
14 )
15
16 print ('Mastodon version: ' + mastodon.retrieve_mastodon_version())
```

You only have to run this once. It registers the `posttoots` application (line 6) with `https://your.mastodon.server` (line 7 – insert the server hosting your account here), sets its scope to `'read'` and `'write'` so you can both post to an account and also read information off the server (line 8), and grabs some credentials and stores them in a hidden file called `.secrets` (line 9).

Although it does not affect the registration, you can tack on lines 12 through 16. Lines 12, 13, and 14 check to see if the app can be activated using the downloaded credentials in `.secrets` and line 16 tells you what version of Mastodon your server is using. This can be useful if your program relies on features available on the very latest Mastodon, but your server has not updated yet.

You only need to run `posttoots_register.py` once as you can use the credentials stored in `.secrets` over and over. `.secrets` (Listing 2) contains a `client_id` (line 1) and a `client_secret` (line 2) that you will be able to use to identify your application each time it has to interact with the Mastodon instance. On line 3 you also have the address of the instance the app is registered with, which is convenient as you will see later on.

So, posting is very straightforward. Consider Listing 3.

After pulling in `sys` (to process the command-line arguments – line 3) and `Mastodon` (for everything else – line 4), you activate your app with the Mastodon server (lines 6 through 8). You could insert here the `client_id`, `client_secret`, and the host, but we have all that in the `.secrets` file, remember? So all you have to do is point to that (line 7).

Then you need to log in with your username and password so the application posts to a specific

Listing 2: .secrets

```
01 53eM1n9lyHR4nd0mUnUMB3r5vAN6G133TeR5X4g
02 KM0r340FdTh3p54M3A60nTnUkKN0WC0hweMigLcng6U
03 https://your.mastodon.server
```

Listing 3: posttoots.py (Basic)

```
01 #!/usr/bin/env python
02
03 import sys
04 from mastodon import Mastodon
05
06 mastodon = Mastodon (
07     client_id = '.secrets',
08 )
09
10 mastodon.access_token = mastodon.log_in (
11     username = sys.argv[1],
12     password = sys.argv[2],
13     scopes = ['read', 'write']
14 )
15
16 mastodon.status_post (sys.argv[3])
```

account (lines 10 through 14). Pass your `username` and `password` to the application and the value of `scopes`. I am not sure why you have to add the latter, but if you don't, the login fails. The value of `scopes` must be the same as what you used when registering your application in Listing 1. Finally, on line 16, you post a message using `Mastodon.py`'s `status_post()` function.

You can use *posttoots.py* like this:

```
./posttoots.py youremail@here.com ⮐
  "Your awesome password" "Tooting from ⮐
  the command line with my own app!"
```

And "*Tooting from the command line with my own app!*" will pop up in your feed (Figure 1).

That's it. The end. See you next issue…

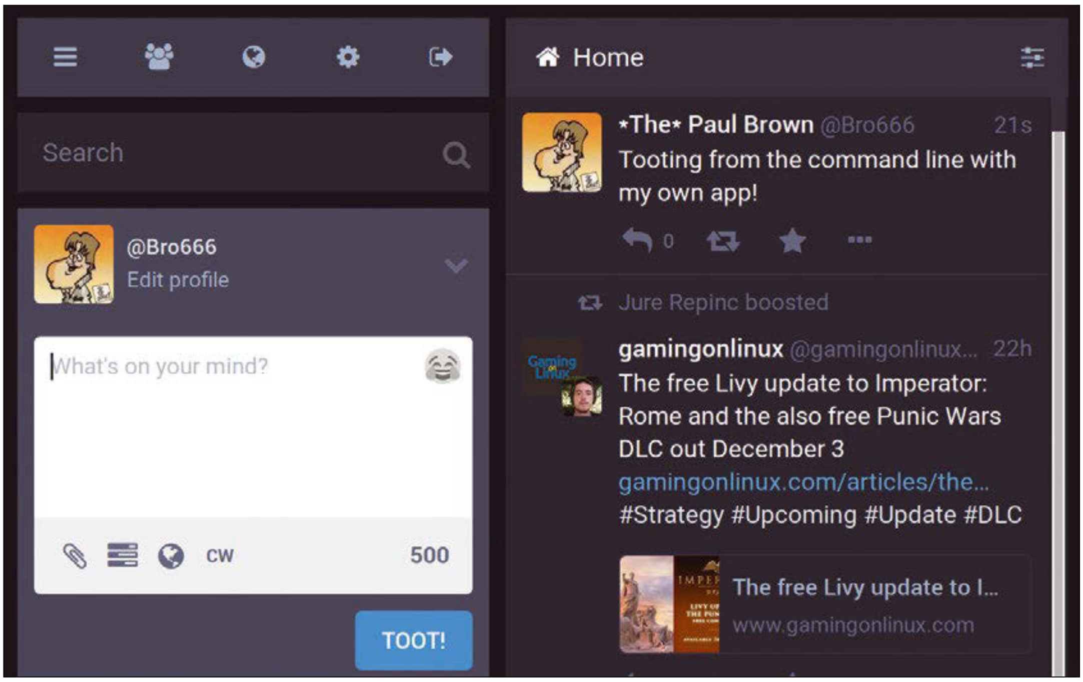
Beyond Simple

Well, not really, because there is so much more you can do when posting. You can, for example, append an image or a video, schedule your toots so they are posted automatically at a later date (easier said than done), set the visibility of each toot, and quite a bit more. To see how all of the above is achieved, consider Listing 4.

You can divide the program into 5 distinct sections:

1. Line 1 sets the interpreter for the program.
2. Lines 3 through 6 pull in the modules you'll need. The `datetime` module lets you convert strings to dates. You will need this when you want to parse dates passed on by the user to schedule toots. The `cv2` module is the Python implementation of OpenCV [3], the library that will allow you to capture an image from your webcam. As you need to parse quite a few parameters to make life easier for the user, let's use `optparse` too. Finally, we import the `mastodon` module.
3. Lines 8 through 41 is the class that does all the dirty work of activating the app, posting, and scheduling.
4. Lines 44 through 53 collect and allocate all the options the user inputs from the command line.
5. Finally, lines 55 through 76 call the functions we need depending on the input from the user.

To go into detail, let's see how we can post an image within a toot.



The Paul Brown @Bro666 21s
Tooting from the command line with my own app!

Jure Repinc boosted
gamingonlinux @gamingonlinux... 22h
The free Livy update to Imperator: Rome and the also free Punic Wars DLC out December 3
gamingonlinux.com/articles/the...
#Strategy #Upcoming #Update #DLC

The free Livy update to I...
www.gamingonlinux.com

Figure 1: Tooting plain text to your account requires a simple Python program.

Posting Pictures

One nice feature that comes bundled with `optparse` is how it provides help for the user. Run this:

```
./ postoots.py -h
```

And you'll get a summary of the possible options and hints as to how to use them.

To start with, `-h` tells you to input your username (the email associated with your Mastodon account) with the `-u` option and your password with the `-p` option.

The `-a` option tells your app what you want to do. You can choose between `post` (send a toot), `schedule` (schedule a toot for later), or `list` (list all the scheduled toots). As you want to send a toot with an image, use `post` here.

Listing 4: posttoots.py (Advanced)

```
#!/usr/bin/env python

import datetime
import cv2
from optparse import OptionParser
from mastodon import Mastodon

class post:
    def __init__ (self, vUser, vPassword):
        f_read = open(".secrets", "r")
        instance = f_read.readlines()[-1].strip(" \n")
        f_read.close()

        self.mastodon = Mastodon(
            client_id = '.secrets',
            api_base_url = instance
        )

        self.mastodon.access_token = self.mastodon.log_in(
            username = vUser,
            password = vPassword,
            scopes = ['read', 'write'],
            to_file = '.token'
        )

        self.post_date = None

    def send (self, vToot, vMedia, vVis):
        self.mastodon.status_post
          (vToot, media_ids = vMedia, scheduled_at = self.
           post_date, visibility = vVis)

    def add_Media (self, vMedia):
        return (self.mastodon.media_post (vMedia))

    def set_date (self, vDate):
        try:
            self.post_date = datetime.datetime.
              strptime(vDate, '%b %d %Y %H:%M %Z')
        except:
            self.post_date = None

    def show_scheduled (self):
        print (self.mastodon.scheduled_statuses())

if __name__ == '__main__':
    parser = OptionParser()
    parser.add_option
      ('-a', '--action',    help = '"post", "schedule" or
       "list"',  dest = 'vAction',  default = 'list')
    parser.add_option
      ('-u', '--username',  help = 'user\'s email',
       dest = 'vUser')
    parser.add_option
      ('-p', '--password',  help = 'user\'s password',
       dest = 'vPassword')
    parser.add_option
      ('-t', '--toot',      help = 'toot text',
       dest = 'vToot')
    parser.add_option
      ('-m', '--media',     help = 'media ("cam" or image,
       video, etc.)', dest = 'vMedia', default = None)
    parser.add_option
      ('-w', '--when',      help = 'when to publish toot.
       Format: "mmm dd YYYY HH:MM Z"', dest = 'vWhen',
       default = None)
    parser.add_option
      ('-v', '--visibility',help = '"public", "unlisted",
        "private" or "direct"', dest = 'vVis', default =
        'public')

    (options, args) = parser.parse_args()

    p=post(options.vUser, options.vPassword)

    mmedia = options.vMedia
    if mmedia != None:
        if mmedia == "cam":
            camera=cv2.VideoCapture(0)
            return_value, image =camera.read()
            cv2.imwrite("pic.png", image)
            del(camera)
            mmedia="pic.png"

        mmedia = p.add_Media(mmedia)

    if options.vAction == 'post':
        p.send (options.vToot, mmedia, options.vVis)

    elif options.vAction == 'schedule':
        p.set_date (options.vWhen)
        p.send (options.vToot, mmedia, options.vVis)

    else:
        p.show_scheduled()
```

The `-t` option contains the text for the toot, and `-m` contains the path and name of the media (picture) you want to send. Given all of the above, your complete command line will look like this:

```
./posttoots.py -u youremail@here.com ➘
  -p "Your awesome password" -a post -t ➘
  "Tooting an image from the command line ➘
  with my own app!" -m /path/to/image.png
```

Looking at Listing 4, once you hit Enter, the pairs of options and values from the command-line arguments get dumped into `(options, args)` (line 53), and the Mastodon object is initiated with your username and password (line 55). The initialization function (lines 9 to 26), takes the URL of the Mastodon instance, `client_id` and `client_secret` from the `.secrets` file (lines 10 to 17), logs the application in (lines 19 to 24), and sets the `post_date` attribute to `None` (line 26).

You will have noticed that the program gets the URL of the Mastodon instance in a very roundabout way. Instead of just loading all the values in from `.secrets`, including the URL, into `client_id`, we do this convoluted thing of opening the `.secrets` file (line 10), dumping the contents of the last line into `instance` (line 11), and then closing the file (line 12).

The reason for this is that there seems to be a bug in `Mastodon.py` that appears when you try to use scheduling: If you rely on a file to load in your instance's URL, anything to do with scheduling will fail with an error complaining about the version of Mastodon running on the server, or some such nonsense. The workaround is setting the URL explicitly on line 16.

This is not the only problem with scheduling as we will see later, but let's get back down into the main body of the program where, on line 58, you check to see if the user has given a path and filename of a media file. If they have, you add the media file to the toot object (line 66 and then lines 31 and 32).

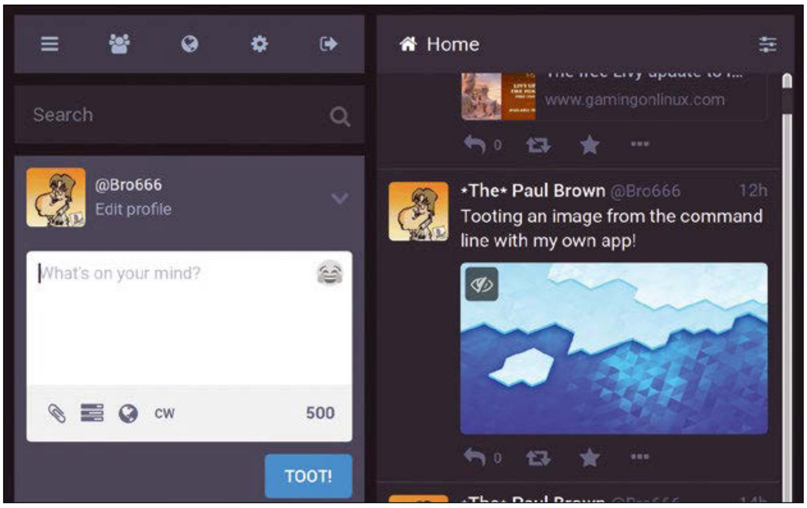
Finally the toot, along with the media, is sent on line 69 by calling the actual Mastodon sending function on lines 28 and 29 (Figure 2).

You can add up to four images to a toot, as well as videos and other kinds of files.

Private Post

Notice that, if using `cam` with the `-m` argument (line 59), you have a block of code (lines 60 to 64) that uses methods from the `cv2` module to grab a picture from the default webcam and send that to Mastodon.

Remembering that you pulled in the OpenCV module (`cv2`) on line 4, you then create a `VideoCapture` object from the default camera on line 60, read what the camera is seeing on line 61, write it out to a file on line 62, and free up the camera



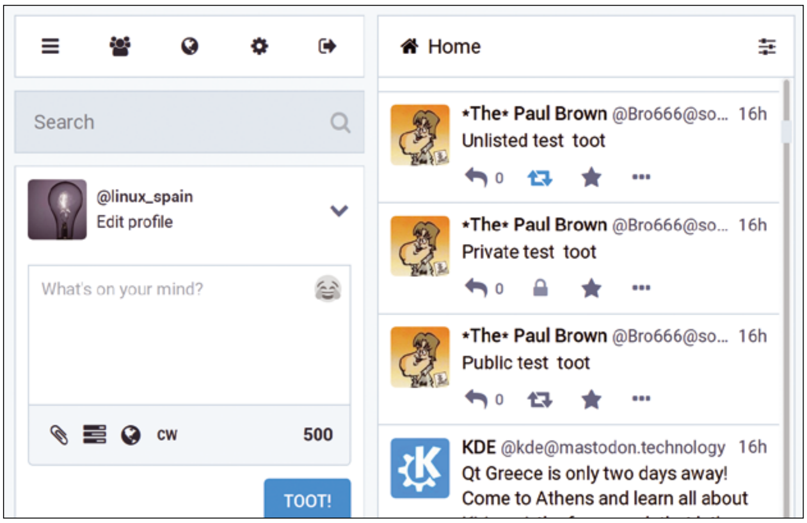
The Paul Brown @Bro666 12h
Tooting an image from the command line with my own app!

Figure 2: You can also send images and videos to your feed relatively easily.

again when done (line 63). All that's left to do is to dump the location of the image into the `mmedia` variable for posting (line 64).

Posting an image from a webcam may be useful for, say, monitoring your home, but it does raise serious privacy concerns if the posts are public. This is where a toot's visibility options come into play. On line 51 of Listing 4, you can see that a user can set a toot's visibility. The options available are:

- **Public** means the toot will be visible to everyone. A public toot can also be found when running a search across the Mastodon federated network.
- **Unlisted** means the toot is visible, but that it cannot be found via a search. Your followers can see it, but a non-follower would need a direct link to the toot to be able to read it. An unlisted toot can be boosted ("retooted"), but the boosted toot remains unlisted.
- **Private** is a bit confusing, because it may not mean what you think. Your followers can see private toots, but non-followers can't. So, in fact,



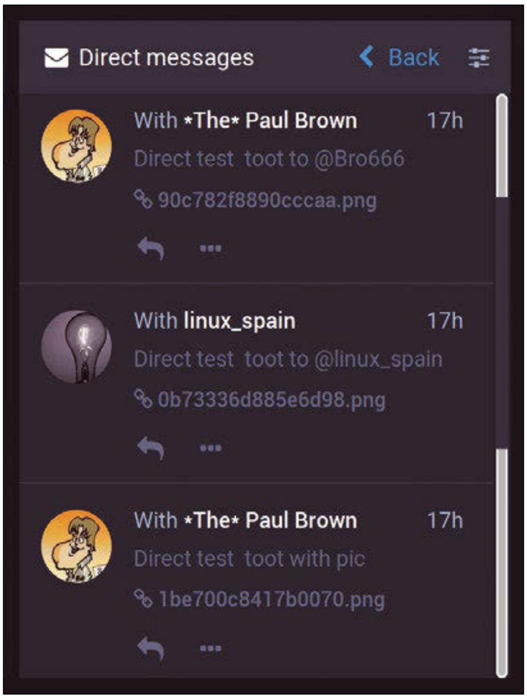
The Paul Brown @Bro666@so... 16h
Unlisted test toot

The Paul Brown @Bro666@so... 16h
Private test toot

The Paul Brown @Bro666@so... 16h
Public test toot

KDE @kde@mastodon.technology 16h
Qt Greece is only two days away!
Come to Athens and learn all about

Figure 3: Three toots with different levels of privacy.



Direct messages

Back

With *The* Paul Brown 17h
Direct test toot to @Bro666
90c782f8890cccaa.png

With linux_spain 17h
Direct test toot to @linux_spain
0b73336d885e6d98.png

With *The* Paul Brown 17h
Direct test toot with pic
1be700c8417b0070.png

Figure 4: Sending a direct message is your best bet for total privacy.

it's not very private at all. However, you cannot boost another user's private toot.

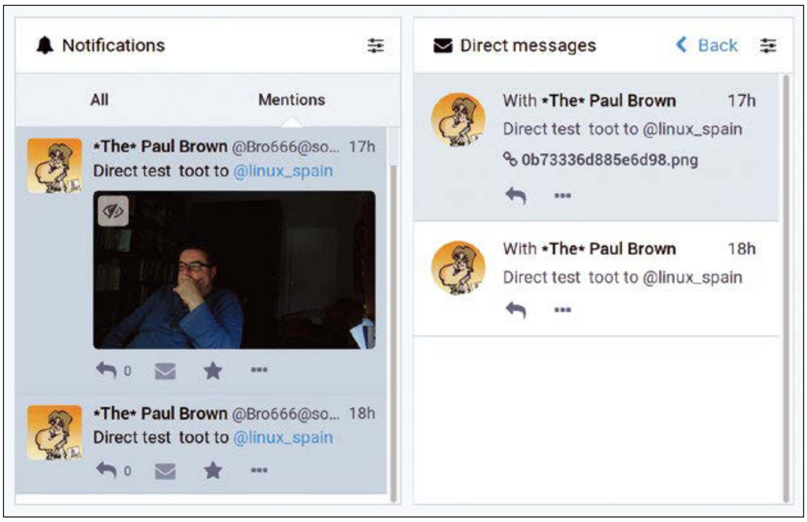
Figure 3 shows three toots with different levels of privacy. Notice the padlock and the absence of a boosting icon in the private toot.

There is a fourth level of privacy which, it turns out, is exactly what you need to keep the pictures from your webcam truly private:

- **Direct** makes the toot a direct message and these are truly private, since they land in your inbox and only you and the owners of the accounts mentioned in the message can see them.

So if you send a direct message with a snap attached but no username (Figure 4, bottom toot) or your own username in the text of the message (Figure 4, top toot), they will land discretely in your Mastodon inbox and won't be seen by anyone else:

```
./posttoots.py -u youremail@here.com ↩
  -p "Your awesome password" -a post -t ↩
  "Direct test toot with pic" -m cam -v direct
```



Notifications

All

Mentions

The Paul Brown @Bro666@so... 17h
Direct test toot to @linux_spain

0

The Paul Brown @Bro666@so... 18h
Direct test toot to @linux_spain

0

Direct messages

Back

With *The* Paul Brown 17h
Direct test toot to @linux_spain
0b73336d885e6d98.png

With *The* Paul Brown 18h
Direct test toot to @linux_spain

Figure 5: Direct messages to your account show up in your feed, but only you can see them.

That said, you will have to click on a link to see the image. But, if you have two separate Mastodon accounts and send a direct message from one to the other, the toot will show up in the recipient's feed with a preview of the image, like you can see in Figure 5.

```
./posttoots.py -u youremail@here.com ↩
  -p "Your awesome password" -a post -t ↩
  "Direct test toot to @your_other@account" ↩
  -m cam -v direct
```

The darker background indicates that it is a direct message and not a regular toot.

Post-posting

Warning: Scheduling is finicky and seems to be supported on only a few Mastodon instances. But, if one day scheduling on Mastodon becomes universally available, this is how you would do it:

Set the `-a` option to `schedule` and set the `-w` argument. The `-w` argument takes a string that contains a date. The format for the date is "*mmm dd YYYY HH:MM Z*", where *mmm* is a three letter abbreviation of the month in lower case (*jan*, *feb*, *mar*, etc.), *dd* is the day of the month (*01* to *31*), *YYYY* is the year (*2019*, *2020*, etc.), *HH* is the hour (*00* to *23*), *MM* are the minutes (*00* to *60*), and *Z* is the time zone (*UTC*, *CET*, *PDT*, etc.).

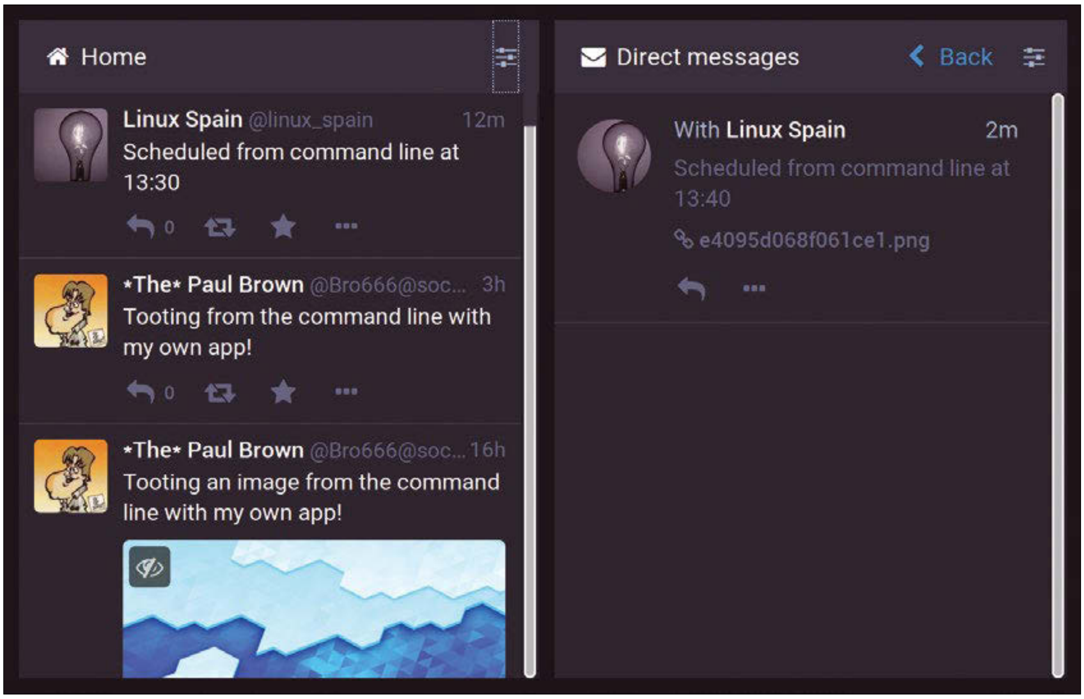
To send a toot at noon on the 31st of January 2020, for example, you'd run:

```
./posttoots.py -w "jan 31 2020 12:00 CET" ↩
  -a schedule  -u youremail@here.com  -p ↩
  "Your awesome password" -t ↩
  "Scheduled from command line at 12:00"
```

In Listing 4, lines 71 to 73 are the ones that deal with the `schedule` option. First, the scheduled date is processed on lines 34 to 38 using Python's `datetime` module if it is a viable date (lines 35 and 36). However, the string will be discarded if `datetime` cannot identify it as a date (lines 37 and 38). Then the toot is sent as usual (Figure 6).

You can see the list of scheduled toots by using the `list` option with the `-a` argument:

```
./posttoots.py -a list -u ↩
  youremail@here.com -p "Your awesome password"
```



Linux Spain @linux_spain 12m
Scheduled from command line at 13:30

The Paul Brown @Bro666@soc... 3h
Tooting from the command line with my own app!

The Paul Brown @Bro666@soc... 16h
Tooting an image from the command line with my own app!

With Linux Spain 2m
Scheduled from command line at 13:40
e4095d068f061ce1.png

Figure 6: Scheduled toots show up in your timeline at the preprogrammed time. You can also send them as direct messages.

Scheduling is cool, but as mentioned above, fickle. As noted elsewhere, if you rely on your `.secrets` file to supply the address of the Mastodon instance you are using, for some reason, nothing related to scheduling will work, and your app will bomb out complaining about the version of the instance. That's the reason for the ugly hack on lines 10 through 12 in Listing 4.

And then there's the fact that not all Mastodon instances support scheduling, so don't be surprised if the above does not work for you. If you try to schedule a post from your command line and then it doesn't show up in your feed at the designated time, it is a pretty sure indication that the server does not support this feature. If scheduling toots is important for you, it will be time to search for another server.

You can try *https://mastodon.social/*. This is the server maintained by Eugen Rochko, one of the creators of Mastodon and the person who implemented scheduling. Scheduling works a treat on this instance.

Conclusion

This is the end of the series dedicated to Mastodon. As you may have gathered, Mastodon is pretty great, and it is easy to see why it has become the most popular service in the Fediverse. It is more flexible than Twitter and much easier to program for. At this point, you should have enough knowledge to do all sorts of things, from creating clients to building bots and becoming a nuisance.

But that does not mean we are done. We still have to explore all the other services, the increasingly popular PeerTube, a contender for YouTube; Pixelfed, which wants to become the Instagram of the federated social media; and so many more. We'll be tackling these and exploring how to best leverage them in upcoming issues.

See you in the Fediverse! ■■■

Info

[1] "Build your own Mastodon client," by Paul Brown, *Linux Magazine*, issue 230, January 2020, pp. 92-94

[2] Mastodon.py documentation: *https://mastodonpy.readthedocs.io/en/stable/index.html*

[3] OpenCV: *https://opencv.org/*

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*.

FOSDEM '20

Date: February 1-2, 2020

Location: Brussels, Belgium

Website: *https://fosdem.org/2020/*

FOSDEM is a free and non-commercial event organized by the community for the community to promote the widespread use of free and open source software. Join thousands of developers from all over the world for this two-day event.

SCaLE 18x

Date: March 5-8, 2020

Location: Pasadena, California

Website: *https://www.socallinuxexpo.org/scale/18x*

SCaLE is the largest community-run, open source and free software conference in North America. It is held annually in the greater Los Angeles area. SCaLE 18x expects to host 150 exhibitors, along with nearly 130 sessions, tutorials, and special events.

AI Hardware Summit Europe

Date: March 10-11, 2020

Location: Munich, Germany

Website: *https://www.aihardwaresummiteu.com/events/ai-hardware-summit-europe*

After three successful events (two in the US, one in China) we're delighted to be bringing the AI Hardware Summit to Munich. The AI Hardware Summit Europe will bring together key players across the AI hardware ecosystem to promote innovation and adoption of systems & silicon for processing AI.

Events

FOSDEM 2020	February 1-2	Brussels, Belgium	https://fosdem.org/2020/
Kubernetes Forum Bengaluru	February 17-18	Bengaluru, India	https://events19.linuxfoundation.org/events/kubernetes-forum-bengaluru-2020/
Kubernetes Forum Delhi	February 20-21	Delhi, India	https://events19.linuxfoundation.org/events/kubernetes-forum-delhi-2020/
Software Architecture Conference	February 23-26	New York, New York	https://conferences.oreilly.com/software-architecture/sa-ny
SCaLE 18x	March 5-8	Pasadena, California	https://www.socallinuxexpo.org/scale/18x
AI Hardware Summit	March 10-11	Munich, Germany	https://www.aihardwaresummiteu.com/events/ai-hardware-summit-europe
Cloud Expo Europe	March 11-12	London, United Kingdom	https://www.cloudexpoeurope.com/
CloudFest 2020	March 14-19	Europa-Park, Germany	https://www.cloudfest.com/
Artificial Intelligence Conference	March 15-18	San Jose, California	https://conferences.oreilly.com/artificial-intelligence/ai-ca
Strata Data Conference	March 15-18	San Jose, California	https://conferences.oreilly.com/artificial-intelligence/ai-ca
SUSECON 2020	March 23-27	Dublin, Ireland	https://www.susecon.com/
Kubecon + Cloudnativecon Europe 2020	March 30-April 2	Amsterdam, Netherlands	https://events.linuxfoundation.org/kubecon-cloudnativecon-europe/
Open Networking & Edge Summit North America	April 20-21	Los Angeles, California	https://events.linuxfoundation.org/open-networking-edge-summit-north-america/
Strata Data Conference	April 20-23	London, United Kingdom	https://conferences.oreilly.com/strata-data-ai/stai-eu
DevOpsCon	April 21-24	London, United Kingdom	https://devops.jaxlondon.com/

Images © Alex White, 123RF.com

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.

Additional information is available at:
http://www.linux-magazine.com/contact/write_for_us.

NOW PRINTED ON recycled paper from 100% post-consumer waste; no chlorine bleach is used in the production process.

Authors

Erik Bärwaldt	32, 56
Paul Brown	90
Zack Brown	12
Bruce Byfield	36, 62
Joe Casad	3, 18
Mark Crutch	71
Jon "maddog" Hall	72
Werner Hein	50
Richard Ibbotson	73
Charly Kühnast	39
Christoph Langner	18, 24, 40
Thomas Leichtenstern	32
Michael McCallister	74
Vincent Mealing	71
Pete Metcalfe	66
Graham Morrison	82
Mike Schilli	44
Tim Schürmann	78
Mayank Sharma	15
Ferdinand Thommes	28
Jack Wallen	8

Contact Info

Editor in Chief
Joe Casad, jcasad@linux-magazine.com
Copy Editors
Amy Pettle, Megan Phelps
News Editor
Jack Wallen
Editor Emerita Nomadica
Rita L Sooby
Managing Editor
Lori White
Localization & Translation
Ian Travis
Layout
Dena Friesen, Lori White
Cover Design
Lori White
Cover Image
© Vlasta Sharavatova, elenasz, Liu Ming, Popa Silvia, and dvarg 123RF.com
Advertising
Brian Osborn, bosborn@linuxnewmedia.com
phone +49 89 3090 5128
Marketing Communications
Gwen Clark, gclark@linuxnewmedia.com
Linux New Media USA, LLC
2721 W 6th St, Ste D
Lawrence, KS 66049 USA
Publisher
Brian Osborn
Customer Service / Subscription
For USA and Canada:
Email: cs@linuxpromagazine.com
Phone: 1-866-247-2802
(Toll Free from the US and Canada)

For all other countries:
Email: subs@linux-magazine.com

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.

Printed in Nuremberg, Germany by hofmann infocom GmbH on recycled paper from 100% post-consumer waste; no chlorine bleach is used in the production process.

Distributed by Seymour Distribution Ltd, United Kingdom

LINUX PRO MAGAZINE (ISSN 1752-9050) is published monthly by Linux New Media USA, LLC, 2721 W 6th St, Ste D, Lawrence, KS, 66049, USA. Periodicals Postage paid at Lawrence, KS and additional mailing offices. Ride-Along Enclosed. POSTMASTER: Please send address changes to Linux Pro Magazine, 2721 W 6th St, Ste D, Lawrence, KS 66049, USA.

Published monthly in Europe as Linux Magazine (ISSN 1471-5678) by: Sparkhaus Media GmbH, Zieblandstr. 1, 80799 Munich, Germany.

NEXT MONTH

Issue 232

Approximate

UK / Europe	Feb 08
USA / Canada	Mar 06
Australia	Apr 06

On Sale Date

Issue 232 / March 2020

Ad Blockers

Many users are familiar with the steps for adding an ad blocker to a web browser, but tools like Pi-Hole and Privoxy let you block *all* the ads entering your LAN before they even reach the network. Next month we explore the benefits of a web proxy and look at techniques for blocking ads at the network level.

Lead Image © enciktep, 123RF.com

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: ***https://bit.ly/Linux-Update***

RYZEN
AMD