FREE DVD

9 Raspberry Pi Projects

QGIS AND QMAPSHACK
Tools for creating and customizing maps

LINUX PRO MAGAZINE
AUGUST 2018

MAPPING TOOLS

Create and edit digital maps with QGIS and QMapShack

4 Open Source Microblogging Tools

Kubuntu Up Close
Explore the new 18.04 release

MakerSpace
9 Handpicked Pi projects

Caddy
HTTPS server out of the box

ODROID-C2
This single-board system is twice as fast as the competition

Ogg Vorbis
Free format for audio files

A Python Script
That solves the Chinese ring puzzle

LINUXVOICE

- Local music streaming with Cantata
- maddog – Adding hardware to FOSS
- Gnome Recipes – Organize your recipe collection

FOSSPicks
- Gimp 2.10
- cointop
- Font Finder
- WireGuard

Tutorials
- odfgrep
- Kdenlive and ImageMagick

Issue 213
Aug 2018
US$ 15.99
CAN$ 17.99

HETZNER ONLINE

to the new Hetzner Online
DATA CENTER PARK HELSINKI

Dedicated Root Server PX61-NVMe

- ✔ Intel® Xeon® E3-1275 v5 Quad-Core Skylake
- ✔ Incl. Hyper-Threading-Technology
- ✔ 64 GB DDR4 ECC RAM
- ✔ 2 x 512 GB NVMe Gen3 x4 SSD
- ✔ 100 GB Backup Space
- ✔ 30 TB Traffic*
- ✔ Location Finland

Setup Fee $148

monthly $ **67**

* There are no charges for overage. We will permanently restrict the connection speed if more than 30 TB/month are used. Optionally, the limit can be permanently cancelled by committing to pay $1.23 per additional TB used.

Choose FINLAND for your server location and SAVE

Thanks to the low electricity prices and the cool climate, this new data center location means that you'll get even lower prices with the same quality infrastructure and service. In addition, the new location will give you even better network coverage and geo-redundancy for your systems. Plus it already meets the requirements of the new European Union General Data Protection Regulation.

www.hetzner.com

Now in Finland

TECHNICAL NAIVETÉ

Dear Reader,

I have used this space in the past to highlight a certain tendency in our culture to overstate the power, stability, and safety of our emerging technologies. For lack of a better term, I'll refer to this trait as technical naiveté. This tendency does not stem from a lack of technical knowledge – in fact, many of the people who suffer from this malady have lots and lots of technical knowledge – maybe even *too much*. The real problem is a lack of awareness about everything else. The hope is that, if you imagine a new technology in a sincere and earnest way, and you have the best of intentions for what you are going to do with it, the world will play along. One views the new technology as if it were in the pages of a science fiction book: A central feature that the story is then draped around, and the reality of the story is gently shaped to reflect the grandeur and importance of the technology.

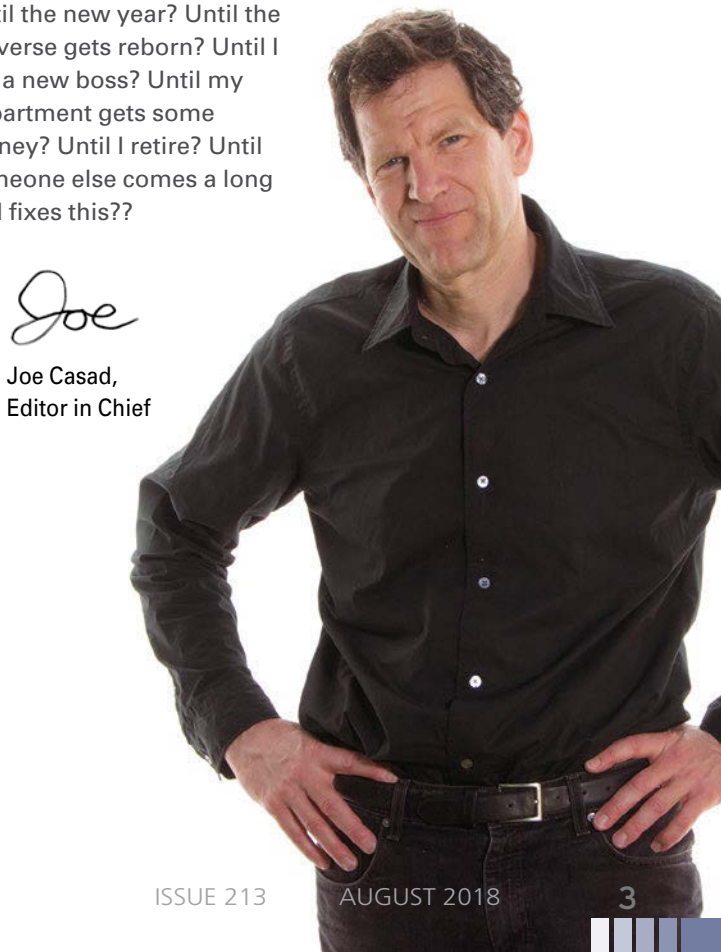
People often write these stories in their heads when they behold new technologies that we invent here in our real world, but reality is a little messier than the elegant spaces of a science fiction novel. That's where we get the debacle of an autonomous mall robot that falls into a fountain and shorts out or a social media system that is supposed to bring us all together but actually incites weird mob behavior. Sometimes it is just a matter of time until we fix the problems and make the technology into what we thought it was in the first place. Other times, we just keep on living in the story that glorifies our technology and understates the danger, because, well, reality is often a little more difficult and considerably more expensive.

I thought I was a time traveler when I read the story at the Reg [1] about a presentation by some security researchers on the state of the Supervisory Control and Data Acquisition (SCADA) infrastructure. The SCADA standard [2] is used for remote control systems in industrial environments such as factories and power plants. Worries about SCADA first emerged when the Stuxnet worm (allegedly created by US spy agencies) burrowed into Iran and caused 1,000 nuclear centrifuges to self-destruct [3]. "Whoa, those SCADA systems are really vulnerable," people said at the time. Did I mention that was eight years ago?

How is it going now for SCADA? Of course, SCADA systems are running all over the world, and some of those systems are well patrolled and maintained, but according to a report by Mike Godfrey and Matt Carr at the BSides conference in London, many SCADA systems are as unsafe as ever, with hopelessly ancient software, featuring hard-coded passwords and a conspicuous lack of encryption. Incredibly, some real-world SCADA systems are actually running on out-of-date Windows computers, including some that are still using Windows 98. Godfrey and Carr even presented a kind of universal SCADA break-in device: an Arduino microcontroller that you can bring to a SCADA facility and it will automatically begin identifying networks, logging in, and issuing STOP commands to start shutting down industrial systems.

Why are some SCADA systems still vulnerable eight years after Stuxnet? One reason is because it is so expensive and difficult to fix old embedded code. But something else is going on that is all about the simple pleasures of technical naiveté. It is all so empowering and cool to imagine that this whole factory can be managed from a single browser window without ever having to get dirty and change anything. Please let me stay in my Jimmy Neutron dream (or in my company's collective Jimmy Neutron dream) and remain in this futuristic vision of vast power at my fingertips, and just keep this rocket flying until…when?

Until the new year? Until the universe gets reborn? Until I get a new boss? Until my department gets some money? Until I retire? Until someone else comes a long and fixes this??



Joe Casad,
Editor in Chief

Info

[1] "Pwned with 4 Lines of Code: Researchers Warn SCADA Systems are Still Hopelessly Insecure": *https://www.theregister.co.uk/2018/06/18/physically_hacking_scada_infosec/*

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

[3] Stuxnet: *https://en.wikipedia.org/wiki/Stuxnet*

LINUX MAGAZINE

WHAT'S INSIDE

This month's issue looks at Linux applications for viewing and customizing maps. You'll learn how to access and edit geospacial data with QGIS and QMapShack. Also inside:

- **Free Microblogging Software** – You don't have sign away your privacy just to tell your friends about your life (page 34).
- **Caddy** – A useful little web server with an easy path to HTTPS (page 50).

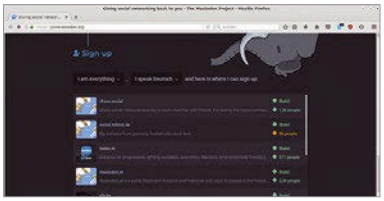
Turn to our MakerSpace section for nine cool Raspberry Pi projects, and visit LinuxVoice for music streaming and recipe collections.

SERVICE

NEWS

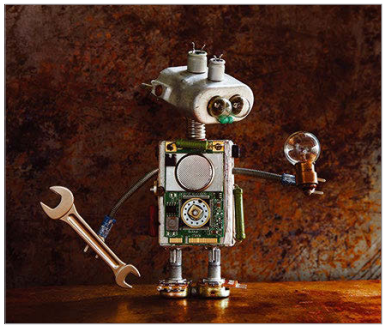
REVIEWS

COVER STORIES



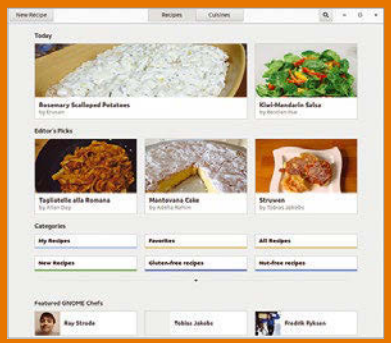
IN-DEPTH

MAKERSPACE



SEE PAGE 6 FOR DETAILS

LINUXVOICE



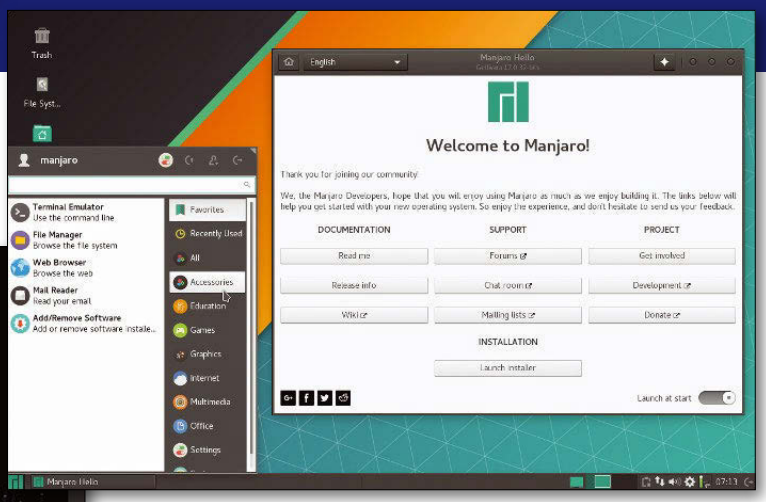
On the DVD

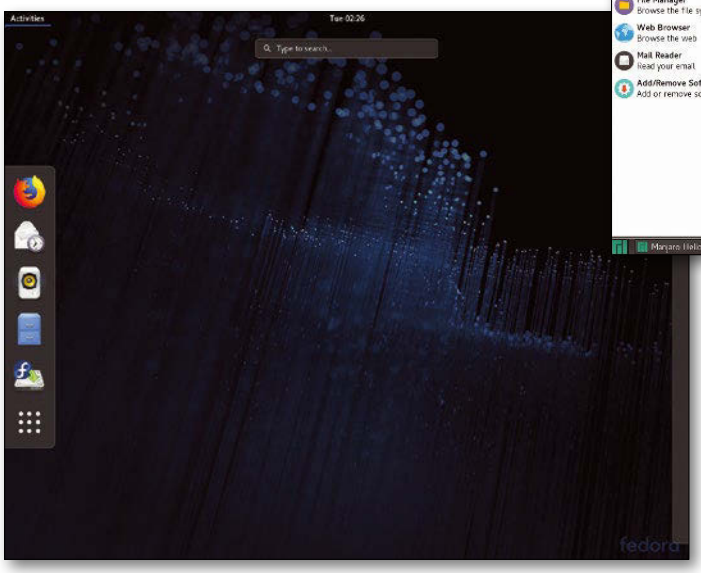
Fedora 28 Workstation

Fedora is a community-based and Red Hat-sponsored distro that strives to keep up with the latest versions of the best Linux tools. The new edition comes with Gnome 3.28, which features improvements to the calendar, contacts, and clocks features, as well as a new tool called Usage that helps diagnose and resolve performance issues. Other improvements include a streamlined installer, improved battery life, and preconfigured access to third-party repositories.

Manjaro XFCE 17.1.10

Manjaro is a fast and user-friendly Linux with intuitive installation and a stable rolling-release development model. Based on the iconic Arch Linux, Manjaro strives to bring the benefits of Arch to the wider world of everyday users. The XFCE desktop included with the DVD combines resource frugality and a full desktop-user experience. The highly regarded Manjaro is undergoing a recent surge in popularity and is currently first on the Page Hit list at distrowatch.com – ahead of both Ubuntu and Linux Mint.





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

Additional Resources

[1] Fedora Project: *https://getfedora.org/*

[2] Fedora Documentation: *https://docs.fedoraproject.org/index.html*

[3] Manjaro: *https://manjaro.org/*

[4] Manjaro Wiki: *https://wiki.manjaro.org/index.php?*

[5] Manjaro Forum: *https://forum.manjaro.org/*

Akademy 2018

Vienna, 11th to 17th of August

Meet the creators of Plasma and Plasma Mobile. Check out bleeding edge versions of your favorite KDE apps. Join the community!

Register for free at:

akademy.kde.org/2018

Hofburg Palace. Vienna

NEWS

Updates on tech s

THIS MONTH'S NEWS

SUSE Forks Red Hat's Spacewalk Project

SUSE and openSUSE developers have forked Red Hat's Spacewalk project to create Uyuni. The fork was announced during the openSUSE Conference in Prague, Czech Republic. Spacewalk is an open-source systems management solution which was upstream for Red Hat Satellite and SUSE Manager.

Klaus Kämpf, SUSE Manager project owner, told us in an interview that Spacewalk was put in maintenance mode, and the project didn't have the developer bandwidth to accept new changes suggested by SUSE.

©Noel Powell, Fotolia.com

Initially SUSE tried to work with Red Hat to hand over the project and Spacewalk trademark to SUSE, but it didn't work out. A fork was the only alternative.

"SUSE Manager's development will be openly available to open source community members for whatever contributions they would like to make to the Uyuni project," Kämpf said.

Uyuni will be an openSUSE/SUSE maintained project, just like openSUSE; anyone can contribute. The current development plans call for releasing a first version this summer. The developers will then decide on a release model together with the community.

openSUSE Leap 15 Announced

In an interview, Richard Brown, openSUSE chairman, said that the focus of openSUSE Leap is a super-stable release of openSUSE targeted for server workloads and professionals who use desktop Linux as a development platform. Leap is based on the latest release of SUSE Linux Enterprise. According to Brown, openSUSE Leap is an enterprise-grade distribution that offers a mix of packages from openSUSE's rolling-release Tumbleweed project and SUSE Linux Enterprise (SLE).

"Having a community distribution that shares a common DNA with enterprise is the smart way to interact with the open source

ecosystem," said Kai Dupke, long-time openSUSE user and senior product manager for SUSE Linux Enterprise 15.

openSUSE Leap 15 makes it easy for users to migrate to SLE, allowing users to gain access to SUSE's commercial support offerings. The latest Leap introduces a new partitioner, integrates the Kopano Groupware, and moves to firewalld. In addition, Leap 15 also introduces a system role selection with a classic "server" or "transactional server" role with transactional updates and a read-only root filesystem. This configuration brings the benefits of atomic updates to the full scope of deployments, from the Internet of Things (IoT) and embedded devices to classical server and desktop roles.

Leap 15 is optimized for cloud usage scenarios as a virtualization guest and offers a great variety of desktops, including KDE and Gnome. Leap also features the return of Live images for simple test driving.

openSUSE Leap comes with KDE's Plasma desktop as the default desktop environment with Gnome as an option during installation.

Download openSUSE Leap 15 for free from the openSUSE website.

SoftMaker FreeOffice 2018 for Linux

There is no dearth of office suites for the Linux desktop, but some professional writers miss some of Microsoft Word's advanced features of Microsoft Word. The German company SoftMaker Software GmbH is trying to fill that gap with its free of cost office suite, FreeOffice.

The company says the new release is completely revised to be fully compatible with Microsoft Office, supporting the Microsoft DOCX, XLSX, and PPTX formats. To help users switch from Microsoft Word to FreeOffice, SoftMaker is offering an optional ribbon interface.

"In our view, the new interface and the improved compatibility with Microsoft Office make FreeOffice 2018 a must-have for 2018," says Martin Kotulla, SoftMaker CEO.

The suite comes with the word processor TextMaker 2018, the spreadsheet application PlanMaker 2018, and the presentation application Presentations 2018.

In a press release, SoftMaker claims, "Aside from seamless DOCX support, the application offers true-to-original handling of numerous other file formats. It opens documents saved in the old Microsoft DOC format just as loss-free as ODT files. The word processor comes with a powerful PDF export, as well as with an EPUB export function, which creates complete ebooks with just a few clicks."

You can download FreeOffice 2018 free of cost at *www.freeoffice.com/en/*.

A New CentOS

CentOS release manager, Karanbir Singh announced the release of CentOS Linux 7 1804, which is based on Red Hat Enterprise Linux (RHEL) 7.5.

CentOS is a community-maintained clone of RHEL, and it is targeted at users who want the functionality of RHEL without the need for Red Hat support. As a result, CentOS is extremely popular among web hosting providers that need thousands of virtual machines to run websites.

As CentOS emerged as a serious threat to RHEL, Red Hat moved swiftly to acquire the project. Many CentOS maintainers joined Red Hat. Since then, CentOS has maintained a measure of independence and continues to be available free of cost.

Although CentOS is seen as downstream of RHEL, in some cases it also works as an upstream source. "Developers and end users looking at inspecting and contributing patches to the CentOS Linux distro will find the code hosted at *git.centos.org* far simpler to work against," wrote Singh.

Users are urged to upgrade to the latest version of CentOS. "This release supersedes all previously released content for CentOS Linux 7, and therefore we highly encourage all

MORE ONLINE

Linux Magazine
www.linux-magazine.com

ADMIN HPC
http://hpc.admin-magazine.com/

pyamgx – Accelerated Python Library
Jeff Layton
Python is quickly becoming one of the most popular languages for scientific computing and is already the most popular language for Deep Learning.

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

AWS Security Scans with Scout2
Chris Binnie
Scout2 is an open source auditing tool that helps you keep your AWS environments secure.

Private Cloud with Microsoft Azure Stack
Florian Frommherz and Michael Lüscher
Azure Stack is an Azure extension that implements an on-premises data center for consistent hybrid cloud deployments.

Monitoring and Service Discovery with Consul • Martin Loschwitz
When dozens of new services and VMs emerge and disappear every day in dynamic cloud environments, conventional monitoring provides false alarms, not operational security.

ADMIN DevOps Focus
http://www.admin-magazine.com/DevOps

Tuning Ansible • Chris Binnie
A number of widely adopted configuration management tools have risen and fallen in popularity since the DevOps craze first appeared.

AWS Automation Documents • Chris Binnie
AWS Systems Manager Automation documents let you customize your Amazon Machine Images to improve security and avoid config drift.

users to upgrade their machines. Information on different upgrade strategies and how to handle stale content is included in the Release Notes," said Singh.

The system upgrade can be performed with these commands:

```
$ sudo yum clean all
$ sudo yum upgrade
$ sudo systemctl reboot
```

Download CentOS at the official download page.

Sonic and Ultrasonic Signals Can Crash Your Hard Drive

Imagine an episode of *Mr. Robot* where Elliot could crash the target hard drive remotely using ultrasonic signals. This story is closer to reality than it seems.

Security researchers from the University of Michigan and Zhejiang University in China have published a paper that demonstrates that the latest hard drives can be crashed using sonic and ultrasonic signals.

Attackers can exploit the vulnerability to destroy hard drives of targets. "Adversaries without special-purpose equipment can cause errors in the hard disk drive using either audible or ultrasonic acoustic waves. Audible waves vibrate the read/write head and platters; ultrasonic waves alter the output of the HDD's shock sensor, intentionally causing the head to park," said researchers.

©Krishna Kumar Sivaraman, 123RF.com

You don't need specialized devices to produce these signals. The sound can be created by the laptop's own speakers.

"Our tests have measured a Dell XPS 15 9550 laptop's output to be as high as 103dB SPL from 1cm away from the laptop. We have observed write blocking using signals as low as 95.6dB SPL. This demonstrates the possibility of using the laptop's own speakers to attack its own hard disk drive," said the researchers.

So, pay attention to sounds in your surroundings.

Microsoft Acquires GitHub

Microsoft has acquired GitHub for $7.5 billion. The acquisition brings financial stability and leadership to GitHub. In addition, Microsoft gets access to 28 million developers who use the platform.

Nat Friedman, the cofounder of Gnome will become the CEO of GitHub. GitHub's current CEO, Chris Wanstrath, will become a Microsoft technical fellow, reporting to executive vice president Scott Guthrie, to work on strategic software initiatives.

"Microsoft is a developer-first company, and by joining forces with GitHub, we strengthen our commitment to developer freedom, openness, and innovation," said Satya Nadella, CEO of Microsoft. "We recognize the community responsibility we take on with this agreement and will do our best work to empower every developer to build, innovate, and solve the world's most pressing challenges."

Microsoft says GitHub will retain its developer-first ethos and will operate independently to provide an open platform for developers in all industries.

According to the press release, developers will be able to use the programming language, tools, and operating system of their choice for their projects — and will still be able to deploy their code to any operating system or cloud platform.

Zack's Kernel News

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

By Zack Brown

Removing Dead Ports

When no one's using a particular piece of hardware, Linus Torvalds becomes very likely to accept a patch removing that hardware from the kernel. It's a similar reason for why he doesn't want to add security patches that don't actually fix security holes, but that just make it "harder" for attackers to design worms. He wants to support hardware that people actually use and fix bugs that actually exist, but he doesn't want to add features that no one will use or that add bloat without actually adding a feature.

Some hardware ports came onto the chopping block recently for this reason. Arnd Bergmann posted a patch to remove Blackfin, CRIS, FRV, M32R, Metag, MN10300, Score, and Tile. He also gave his opinion on why these particular pieces of hardware had disappeared from use. Apparently, as he put it, "while the eight architectures are extremely different, they all suffered the same fate: There was one company in charge of an SoC line, a CPU microarchitecture and a software ecosystem, which was more costly than licensing newer off-the-shelf CPU cores from a third party."

The moral of that story seems to be: Companies shouldn't make custom hardware when generic will do just as well. It's sort of the same reason companies should use Linux instead of writing their own operating system whose key features would soon be incorporated into Linux anyway and be better maintained as well. But apparently this lesson must be learned over and over again.

An interesting aspect of this particular purge was the idea of how best to make it easy for future developers to resurrect those architectures in Linux if they wished. Even when clearing out old and unused parts of the kernel, Linus wants to leave the door open to bring them back if necessary. For that reason, he wanted each architecture to be removed in its own separate patch.

In reality though, as was pointed out in the mailing list discussion, it'd be virtually impossible to resurrect any of these ports by simply reverting the patch that took them out. Geert Uytterhoeven explained that with the kernel itself constantly advancing, anyone trying to bring back one of those ports would probably want to use the newer, more modern features, rather than do a simple patch reversion.

Pavel Machek said it would be more realistic to revert a given patch and then do incremental improvements, in order to get to the place Geert was talking about. Implementing an entire port from scratch, he felt, would not be realistic. Although as Arnd put it, the kernel was developing in ways that would ultimately be incompatible with a pure revert, and very soon a reversion simply wouldn't be possible. And he also added that some of these ports had become so outdated that the kernel probably hadn't been able to run on them for a long time.

The discussion ended, and the patches are certainly going to be accepted. The thing I like about this debate is the attention given to future developers who might care about these architectures that are all but dead. There really is a tremendous focus on ensuring that Linux runs on every piece of hardware that anyone might actually use. And there really is an equally fine attention given to keeping the kernel free of dead architectures and of dead code in general.

New Minimum GCC Version Jumping from 3.2 to 4.8

For many years, GCC v3.2 has been the official minimum. This means that version 3.2, and any later GCC version (with a couple of special exceptions), was officially supported to compile the Linux kernel successfully. Supporting such an ancient minimum is desirable because the kernel people don't want to make assumptions about what software is running on user systems. They want

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**.

anyone to be able to compile a working kernel, using pretty much whatever versions of the tools that are available.

The problem with this is that it isn't true. The topic recently came up on the mailing list, and at one point Linus Torvalds said, "our *documentation* may claim that gcc-3.2 is the minimum supported version, but Arnd pointed out a few months ago that apparently nothing older than 4.1 has actually worked for a longish while, and gcc-4.3 was needed on several architectures."

The question at that point became what version to list as the new official minimum. Clearly it should include as many actively maintained Linux systems as possible. But in order to maximize the ease of kernel maintenance, it should also be the highest GCC version that the Linux developers could get away with. After all, newer GCC versions support programming constructs that are much better than older versions. Having to support those older GCC versions means having to maintain whole branches of ugly kernel code that exist solely for those older GCC versions.

Step by step, the kernel developers walked up the GCC versions, trying to eliminate them from consideration. With 4.3 needed on several architectures, it was a natural starting point. Over the course of discussion, it came out that GCC 4.4 had certain breakage that, although it didn't seem to hit anyone using a regular Linux distribution, was worth taking note of. It was also significant that GCC's `asm goto` feature had first appeared in version 4.5 and allowed the kernel to use much cleaner code than earlier versions. Bumping the official GCC minimum to 4.5 would therefore allow the kernel folks to rip out a lot of ugly code that currently had to exist alongside more recent `asm goto` code. Linus in particular was in favor of doing that. So for awhile, it looked like GCC 4.5 would be the new minimum.

However, Arnd Bergmann pointed out that GCC 4.5 was not able to build kernels for some of the newer ARM systems. In tests he'd done last year, he'd identified GCC 4.6 as a better candidate for the official minimum, for that reason.

But then the question became: Were any Linux distributions – especially the older releases that were in "long-term maintenance mode" – actually using GCC 4.6? Or were they using something more recent? If they were using anything more recent, Linus reasoned, then the official GCC minimum might as well be bumped up to that.

Arnd took a look and found that some of those long-term distributions were using GCC versions older than 4.6, which meant they were not relevant to the question, since Linus was already committed to bumping the GCC minimum up to at least 4.6, and the rest of those distributions were either about to reach their end of life or used GCC 4.8.

Presto! GCC 4.8 would become the new official minimum: new enough to let the kernel use the `asm goto` programming construction, and old enough to cover every Linux distribution still in use. As a result of this decision, it's likely that large swaths of code will be considered dead and will be gutted out of the kernel tree.

Intel Considering Hardware Changes to Mitigate Security Flaws

Folks from Intel are collaborating with the Linux kernel developers to find good ways to fix or work around the Spectre, Meltdown, and other hardware flaws that have been discovered in their chips. Recently, Dave Hansen from Intel asked if a particular hardware change seemed useful to the Linux folks. Specifically, he said, Intel was considering adding a new bit to the `IA32_ARCH_CAPABILITIES` register, to tell when the return stack buffer (RSB) was empty. Normally, the RSB would receive data in a stack and pop it off when requested. But if the stack ever emptied out, it became vulnerable to a Spectre attack. The new bit would check this empty condition, so the attack could be avoided.

Linus Torvalds approved wholeheartedly, since this would let the kernel avoid having to do certain unpleasant tests. At first, there were also various possible objections to Dave's proposal, as various kernel folks analyzed any possible ways for attackers to try to work around Intel's fix. But one by one, these objections were dropped, and the change was generally approved.

Of course, Intel will also be weighing many other considerations before making these sorts of feature changes to their hardware, but it's excellent to see this sort of open collaboration happening.

Enhancing Asymmetric Process Migration

In the before-time, clustering identical CPUs was a big deal. Nowadays, symmetric multiprocessing (SMP) systems are the standard for general-purpose computers. The new weirdness involves asymmetric multiprocessing (AMP). It seems like all the latest person-enhancers want to have certain CPUs for the hard work and certain CPUs for the so-so work.

How does the kernel decide which process to migrate to which CPU when there could be any number of variables to consider? Dietmar Eggemann and Quentin Perret have been working on this issue, along with massive numbers of other developers across the gadget-making industry. They posted a patch recently, targeting systems with energy-efficient CPUs running alongside less efficient hardware. They wanted the system to load up the energy-efficient CPUs with as many processes as possible, before resorting to the less efficient CPUs.

The problem with their code, as they readily acknowledged, was that in order to identify the best available CPU that wasn't already at maximum load, the patch would run through a brute-force algorithm that would only really be efficient on systems with a relatively small number of CPUs. As Dietmar put it, "This patch is an attempt to do something useful, as writing a fast heuristic that performs reasonably well on a broad spectrum of architectures isn't an easy task."

There's definitely some justification for this approach. A lot of gadgets on the market only have a small number of CPUs, so there's a fair chance that a lot of hardware would benefit from this patch. On the other hand, there are various kernel gatekeepers, including Linus Torvalds, who prefer to get patches that solve a whole problem in the most natural way, rather than doing something that's good enough for some users, but misses the main point. So although there was no major objection to this patch, it's likely that someone will have something to say before it makes it into the kernel.

Protecting Users' System Control

Security is sometimes in the eye of the beholder. To the user, a secure system is one that they control themselves. To various vendors, on the other hand, a secure system is one in which the user cannot override the will of the vendor.

This dates back to the early days of "Trusted Computing" and beyond. The argument is that if the vendor can control the entire software chain, they can leverage all sorts of things like proprietary video and audio content, unblockable ads, microcharges for time spent on particular activities, and so on. All of these things become possible, if only you can prevent the user from altering the system.

This was what software companies were trying to do in the 1980s, before Linux came along – force users into a position where they could only use that particular company's offerings. Countering this was not the only motivation behind the arrival of GNU, Linux, and other free software, but it was a big factor.

It's a war that neither side has won. Linux keeps trying to give users control over their systems, and proprietary corporations keep trying to take that control for themselves. Things like the Linux-based Android operating system, with app stores full of closed-source Android apps, are a continuation of that debate.

Recently, David Howells submitted a patch to implement "lockdown," a feature designed "to protect against unauthorized modification of the kernel image." As written, it would operate in conjunction with UEFI Secure Boot, which prevents "unauthorized" operating systems from loading onto a given machine. Matthew Garrett proved to be the patch's biggest exponent, posting many times in its defense.

However, Linus Torvalds, joined by Andy Lutomirski and others, was less interested in the patch's potential benefits and more interested in the ways it might be abused by vendors, to wrest control of a given system away from the user and keep it in the hands of the vendor alone.

A big element of the debate involved accusations by Linus that Matthew was unwilling to admit the true purpose of the patch and kept skirting the real issue in his technical responses.

One of Matthew's clearer defenses of the patch was that the whole point of Secure Boot was to prevent unauthorized kernels from loading into the system. And the point of lockdown was to prevent the running system from being modified. Without lockdown, a user could easily bypass Secure Boot by simply loading a signed kernel that would in turn load any kernel the user desired. If the user could do that, Matthew said, then there would be no value to Secure Boot, because the user could just load whatever kernel they wanted. This kind of one-kernel-loading-another action is called "chainloading."

Another element of Matthew's defense of the patch was that if users didn't like the feature, they could simply disable it in their kernel. They could use a kernel that simply didn't include those security features.

But Linus and Andy, as well as Al Viro, felt that this was a disingenuous argument. As Al put it, this made the assumption that the user had the ability to turn off Secure Boot. If the manufacturer includes firmware that enforces Secure Boot, then Secure Boot itself becomes "a misfeature that has to be worked around. Making that harder might improve the value of [Secure Boot] to said manufacturers, but what's the benefit for everybody else?"

And in a similar vein, Linus said to Matthew, "I may want to know that I'm running *my* kernel, but once that is the case, I trust it. In fact, I tend to trust it more than some random vendor key. You should too." He also added, "secure boot can be hard to turn off. Sometimes 'turn off' means 'you just have to add your own keys'. Yes, on x86 hardware at least at some point MS actually had the rule that it has to be something you can turn off. That rule is apparently not true on ARM, though."

Matthew never gave an inch in the debate – he never affirmed that the goal of the patch was to lock users out of controlling their systems. Ultimately the patch was not accepted. But the whole discussion shows that even today, there is a significant effort to "secure" systems against their proper users. ■■■

Gnome Foundation reaches out to the world

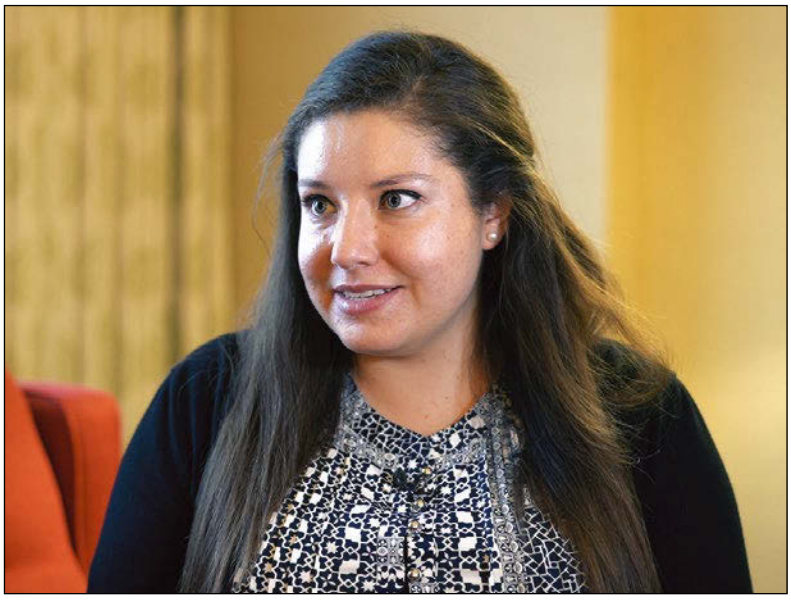
Meet the President

Gnome Foundation president Nuritzi Sanchez is on a quest for a new generation of users.

By Swapnil Bhartiya

Last month, when I was in San Francisco covering the Red Hat Summit, I realized the president of the Gnome Foundation also lived in the area. We connected on WhatsApp and decided to meet for dinner. Nuritzi Sanchez has been involved with open source for more than six years as part of the Endless Computers team. She joined the Gnome Foundation in 2015, and then in 2016, she was elected as the president of the project.

As a newcomer, she finds herself in a unique position. "I have been here only for six years, which is nothing," said Sanchez, "Being a newcomer myself, I can see how important it is to be very welcoming to people who don't have a long history with the project." Gnome, which was founded in 1997, is over 20 years old, and some developers have been around since the beginning. "It's great that we have these veterans who are our North Star regarding what free software should be. You have to have that spectrum," she said. "We also need new users."



As the technology landscape is changing, traditional venues (such as Linux User Groups) that used to bring new users to Linux are disappearing. "I can still identify with many people who haven't really used free software before. I see myself as a bridge, especially for women," she said. "I've noticed that people feel they can approach me easily, and I tend to be a friendly face in general. I have been able to connect with these people."

Sanchez says it's as much about Free Software as it is about making technology accessible to more people and improving the quality of life. One of the reasons she joined the Gnome Foundation is because she believes in the notion of making technology for the whole world. "I love the philosophy behind free and open source software, and when you love something, you internalize it, and it becomes part of your life."

Bringing New Users to Free Software

One of the core focus areas for Sanchez is to bring new users to Linux. As part of the Endless Computers team, which caters to many different countries, she could see that Gnome internationalization was critical to the company's success. If people can't use a software tool in their own language, it won't see adoption. She found that tools that help in internationalization were not very friendly to non-developers. "You have to be familiar with PO files and the general development workflow, but at times, there are opportunities for translators who happen to be teachers or non-techies. These kinds of people find it hard to get involved," she said.

It made Sanchez wonder how a massive, community-driven project like Gnome actually worked. How were decisions made? How were funds and budgets allocated to areas where the project needed to grow? How did they attract new users while at the same time retaining old ones? As she started looking for answers to these questions, she found herself getting more and more involved with the project. Her curiosity eventually led her to

become the president of the Gnome Foundation.

"I started helping in creating policies to further the growth of the Gnome community. While the software development work was under the control of the developers, the foundation was able to have an impact through channelizing funds in the right direction. I see it as a way to empower the community to keep expanding. This year, we're investing more money into the marketing and engagement side of things, where people from the project go out for a hackfest and interact with local communities and bring new people to the project. Paying for pizza or drinks for such events might not be a big deal in the Western society, but there are countries where $50 could be a lot of money for one hackfest. It does make a big difference. The whole point is how to use this money to reach out and bring new people to free and open source software."

When she joined the project, one of the questions she kept hearing was how was she, or the board, going to help expand Gnome and attract new users to the Gnome desktop at a time when web technologies and mobile devices have become shiny new things that everyone wants to chase. "That's something we've been focusing on in the last couple of years. We've been working on different tools for users to find projects and contribute. And we are structuring our policies, at the board level, to help make it easier for more people to join."

It's not just about attracting new people; the foundation organizes events that bring the whole community together to meet in-person and plan for the future. Moving to GitLab (ahead of Microsoft's acquisition of GitHub) was one of the many decisions that came out of such an event. The Gnome Foundation also offers legal support to the Gnome project. It pays for the infrastructure that hosts the code. It hires interns to work on the various projects, and it also signs contracts for specific projects. "We try to provide as much support as we can so that developers can focus on their project instead of worrying about these background tasks," she said.

The Gnome Foundation has two full-time paid employees, including the executive director of the Gnome project; however, the presidency of the Gnome Foundation is an unpaid position; Sanchez does all of this work as a volunteer. "*President* is a legacy term from the days of [Gnome cofounder] Miguel de Icaza," she said. "The president of the board of directors helps direct the board, to make sure that everything is done properly according to the goals. We appoint the executive director to execute those goals into actions, forge relationships, and build the team. We have the power to hire and fire the executive director."

Ubuntu's Impact

Gnome used to be the default desktop for Ubuntu, which is used by millions of users. When Ubuntu switched to Unity in 2011, Gnome lost millions of users and the support of the powerful Ubuntu development community. Last year, however, Canonical decided to shut down the Unity project, and Ubuntu went back to Gnome.

"We're excited about the return of Ubuntu. We are doing our best to welcome Ubuntu users and the community behind the project. Canonical has joined our advisory board, and their teams have started to work with our design teams to better integrate the project," she said. "System76, a Linux-focused hardware vendor, has also joined the advisory board, so a lot of momentum is building up for Gnome. I see a lot of great things happening in 2018 and 2019. Since we also have an executive director now, there are a lot of changes in store for Gnome that you will see in the coming months."

Future Growth

Although the US and Europe might be moving towards smartphones and IoT, many emerging and developing economies are still focused on the Linux desktop. The Gnome Foundation is seeing the emergence of new communities that are becoming active. "We see a lot of interest from countries like Turkey and Kenya. Latin America is also showing some growth. We used to be very strong in those regions, but some of these communities died off. Now we see some revival, which is exciting. I hope that these communities won't just adopt and promote Gnome but will also become active contributors. To make these communities successful, we need to encourage diversity – to create products for our global population."

Diversity is a very complicated term; it could be about language, culture, education, or technical background. "When people don't feel comfortable speaking in a certain language, or a certain tool is blocked in their country, it affects the project. We are thinking about how to address such challenges and facilitate the core development of Gnome. We also want to get them involved with the decision-making process so that they can provide feedback and see the changes that they need," she said.

Conclusion

After talking to Sanchez, I came to realize that most open source projects these days – from OpenStack to Linux Foundation – are focussed on supporting the people who are creating these new technologies. Take care of the developers, and they will take care of the technology. It seems Sanchez has a similar vision to grow the Gnome project, its scope, and its reach by empowering the people behind it. ■■■

Create, display, and evaluate geodata with QGIS

Map Game

Create and display geospatial data with the versatile and popular QGIS. *By Erik Bärwaldt*

Most modern mapping information is not stored as a single complete picture, the way it used to be when you found your geographic information from a map on a wall or in an atlas. Maps have become fully integrated with the digital revolution. The map you view from your computer screen or telephone is transmitted to the device as a bundle of geospatial information that is stored in a file or database. An application on the receiving device assembles this data into a view that the user can recognize as a map.

This geospatial data is available online from many sources, including research institutions, government agencies, and nonprofit groups. Of course, it is also possible to create your own geographical information file, which is the modern equivalent of drawing your own map. Another option is somewhere between the alternatives of downloading or rolling your own: Many users start with public information and then customize the map by adding their own annotations and features.

Linux supports numerous Geographical Information System (GIS) tools that help to create, edit, analyze, and visualize geospacial data. One of the most popular GIS solutions is QGIS [1]. The Quantum-GIS geographical tool, which has undergone continuous development since 2002, is licensed under GPLv2 and available for several platforms.

QGIS uses a PostgreSQL database with the PostGIS extension to capture geographical elements. The modular structure of the software allows the user to extend the functionality by adding plugins. QGIS also supports interfaces to OpenStreetMap, GRASS GIS, and various statistics programs. The system can handle all common vector and raster graphics formats.

With its simple user guidance and sophisticated display functions, QGIS has established itself as an important tool for research professionals. Excellent multilingual

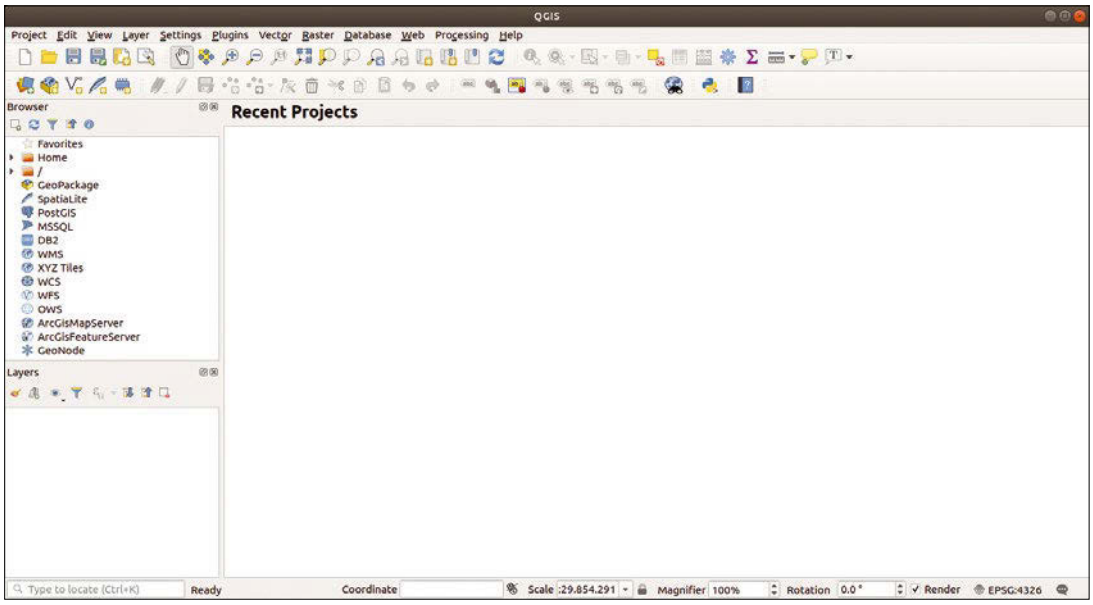


Figure 1: The left panel of the main window hints at the wealth of possibilities offered by QGIS.

documentation and tutorials mean that the software is also quite accessible to beginners.

Installation

You will find precompiled QGIS packages in the repositories of all major Linux distributions; however, if you want to be sure you have the most recent version, manual setup is a better option. The project website provides detailed documentation on integrating QGIS into various systems [2]. Be sure to read the notes on supported operating system versions, because not all program versions work with all Linux distributions due to missing dependencies. For some Debian derivatives, you must first install the *apt-transport-https* package, because the QGIS repositories work with SSL encryption.

To start QGIS, click *QGIS Desktop* in the Start menu. First, a configuration window lets you decide whether to transfer data from a previous version. After you click on one of the options, the program window opens. The window appears largely empty for new installations without imported data (Figure 1). The main window is divided into five sections: At the top, you will find an extensive menubar and a buttonbar. On the left, a browser and a layer area appear below each other, and on the right you will see a list of recently edited projects.

If you load one of your projects, the map view appears in the right window area. The browser and layer areas display the map elements in a tree view. The browser displays data carriers, linked geodatabases, and map services; the layers represent the different levels of a map. The software visualizes the layers hierarchically, so that moving or (de-)activating the individual layers changes the map image accordingly. A status bar at the bottom of the window provides information on coordinates, scale, and magnification of the map.

Individual

The flexible interface can easily adapt to your ideas. For example, minimizing the layer and browser view creates more space for the map. You can place individual window segments anywhere by dragging the titlebar while holding down the left mouse button. To switch the window segments completely on and off, use the dialog panels in the View menu.

The toolbar consists of individual tool groups that you can move individually to the margins of the program window and arrange vertically. If necessary, switch complete tool groups on or off via *View* | *Toolboxes*.

To remove individual unneeded buttons from the toolbars, open the *Settings* | *Interface Customizations…* menu and check the *Enable customizations* option in the overlaying window. You can then navigate through the individual menu hierarchies and deactivate unnecessary options by unchecking the boxes (Figure 2). To apply the changes, restart the software.

For friends of shortcuts, QGIS offers the option of defining individual shortcuts for different functions in the *Settings* | *Shortcuts* menu. In the *Settings* | *Options* dialog, you can also adjust the font and icon size to your requirements. This function is particularly useful for very high-resolution screens, since the default size for icons and fonts on screens beyond FHD resolutions (1920x1080 pixels) often appear too small.

Mess of Data

Many public institutions provide data on a variety of topics for free download over the web. Other geographical data is available

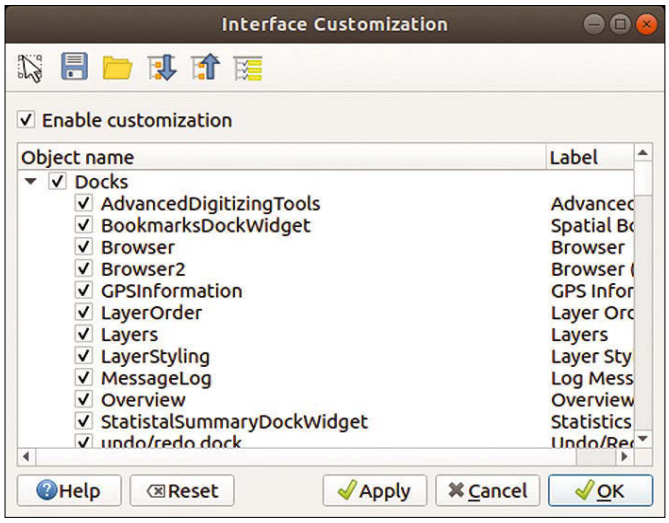


Figure 2: Activate or deactivate the control elements in the Interface Customization dialog box.

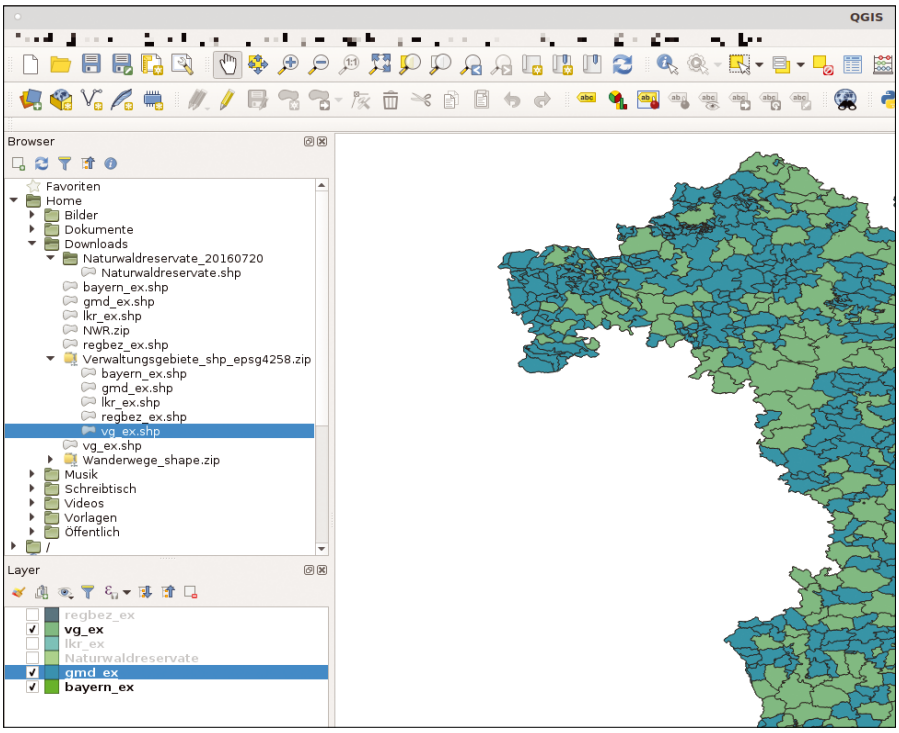


Figure 3: Maps consist of several independent layers, which QGIS lists on the lower-left corner.

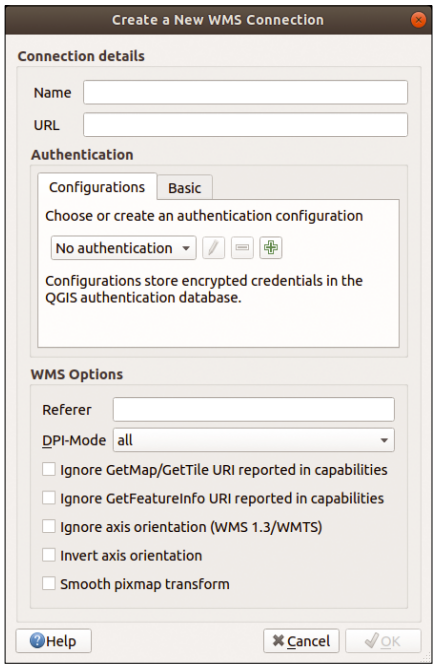


Figure 4: Adding data from a WMS server.

in the form of Web Map Service files (WMS) on the servers of public institutions.

QGIS lets you load this free geographical data into a new or existing project. In the menubar in the *Layer* | *Add Layer* menu, you will find various options for creating a new map layer. Depending on the data format, you have to differentiate between vector and raster data. You can also integrate a database into the software, and you can then localize and load the data in a small file manager on the local mass storage system. The software supports several common database formats.

The downloaded data appears on the left in the layer area.

You can complete this task more elegantly in the browser area, which also contains a small file manager. From the browser area, you can transfer the data using drag-and-drop. If several geo files are in the directory, right-click on the desired file and select *Add Selected Layer(s) to Map* (Figure 3).

Online

If you want to add data from a WMS server to your own project, you need to know its URL. You can usually find the URL on the providers' website and copy it to the clipboard first. Then open the *Add Layer* | *Layer* | *WMS* dialog in QGIS. In the new window, enter the URL in the input field and give the map a meaningful name. A final click on *OK* accepts the data (Figure 4).

You can usually adjust the size of the data provided on the WMS server to your needs by setting an appropriate value in the *Magnification* box below the map area. However, the program reloads the maps from the server each time. Complex projects therefore require a fast Internet connection. The integration of large, high-resolution maps into a project also puts a great strain on the computer, as it has to render very large amounts of data. Depending on the hardware, it

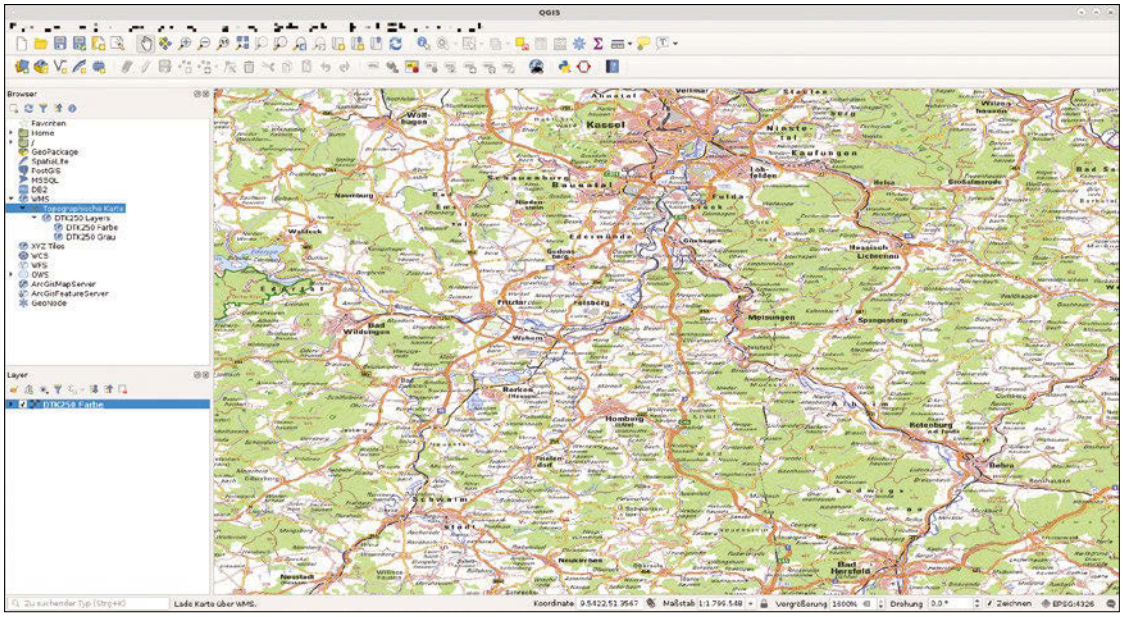


Figure 5: The WMS data is easily integrated and appears as layers for the project.

can take more than ten seconds for a new map to appear.

Geodata that you obtain from a WMS server and integrate into your project cannot be modified. You insert the WMS data like locally stored raster and vector data as layers into your project (Figure 5).

Tile Server Fix

QGIS also allows you to integrate external data from OpenStreetMap and other services. However, the change from version 2.18 "Las Palmas" to the new version 3.0 "Girona" resulted in some problems.

With the older QGIS versions, a plugin named QuickMapServices could be integrated into the program via the dialog *Extensions | Manage and Install Extensions…*, which then offered convenient access to the relevant services via the Web menu. By the way, the same applies to several other addons, which are not yet available for the new version. However, the developers are gradually completing the plugin list.

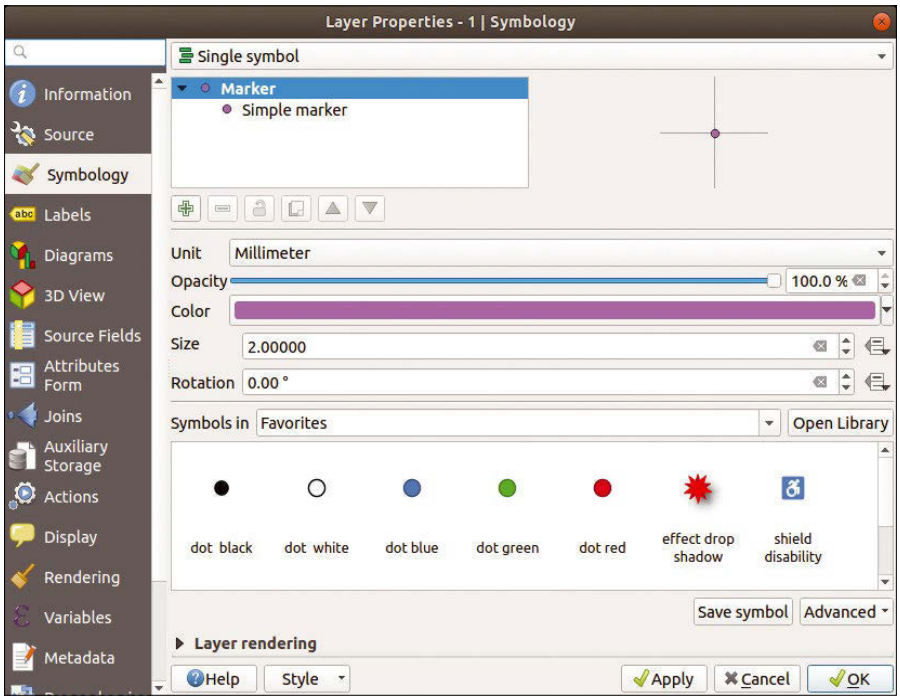


Figure 6: The Layer Properties dialog lets you modify symbols.

Until then, you need to integrate this external data into your installation via tile servers. The tile servers provide tiles with raster data [3]. First you copy the desired URL to the clipboard. Then open the *XYZ Tiles* dialog in the QGIS browser segment by right-clicking on it and then clicking on the *New Connection…* option.

In the dialog that now opens, copy the URL into the corresponding input field. To allow the program to call the URL correctly, remove all *$* symbols from it.

Do-It-Yourself

To insert your own data into maps, generate files in shapefile format (a geospacial vector data format for GIS software) [4]. Then fill the files with the desired raw data. For example, you might wish to add the coordinates of roads, railway lines, waterways, or buildings. The software saves these files as individual layers that can be shown or hidden.

To create a shapefile, select the *Create Shapefile Layer…* option in the *Layer* | *Create Layer* menu. First enter the data type and the coordinate system in the dialog. When entering the coordinate system, make sure that it corresponds to the base plane. Then enter the fields, where you can choose between different types. Note that QGIS does not support non-standard characters.

Then click the *Add to Field List* button to add the new field to the list at the bottom of the window. *OK* closes the window and opens the file manager for saving the shapefile in the older program versions. When the file is saved, the software transfers the file with the arbitrary assigned name to the layer segment.

Then fill the shapefile with the desired data. Select the layer and then activate the editing mode by clicking on the *Switch editing mode* button in the upper-left corner of the program window. Then click on the *Add object* button and select a corresponding object in the map view on the right side of the program window.

A mouse click on the desired object opens an attribute window in which you enter the different properties of the object based on the predefined shape files. Complete the entry by pressing *OK*. The software then transfers the new object to the map view.

Symbolic

QGIS offers powerful tools for displaying data in the individual layers. To make adjustments, double-click on the layer, which opens a window for modifying the geodata. The layer can be a point, line, or polygon layer.

In the *Display* dialog, you will find various customization options for symbols and categories of symbols. QGIS comes with libraries of various ready-made symbols, which you can switch between using *Open Library*. In addition to the actual symbol display, the program also lets you change the color, opacity, size, and alignment (Figure 6).

On saving, QGIS transfers the individual symbol to the active library. In addition, SVG vector files with symbols can be downloaded from the web, integrated into the software, and then modified like the preset symbols.

By categorizing attributes, QGIS also offers the ability to customize individual attributes in a single pass. To do this, define a column of a layer attribute table as a category in the *Layer Properties* settings dialog. Then assign a symbol from the existing libraries to this category so that the tool displays all data with this attribute uniformly using the selected symbol. If necessary, you can modify the symbol's look. If you want to change certain attributes depending on existing values, QGIS can also make rule-based modifications.

Dialogs

Individual input dialogs facilitate data acquisition enormously, especially for large projects that require a lot of human input. QGIS also offers many options for creating and optimizing your own input masks. You can create tabs with groups of input fields, and you can also integrate check boxes, rotary and slider controls, and date fields. The software also lets you define ranges to prevent nonsensical and thus falsifying entries. QGIS does not make you commit to the order of the attributes in the attribute table: You can modify the sequence using drag-and-drop to shift input fields into the desired order.

Objects

To edit individual objects in layers in QGIS, you have to select them by clicking on them. To select several objects simultaneously, capture them by drawing a geometric shape, such as a rectangle or a circle, around the objects you wish to select. For irregularly arranged objects, such as trees in a tree register, choose *Select objects via polygon function*.

If the layer contains many irregularly arranged objects with different properties and you want to select objects with a very specific property, choose the button labeled *Select objects via expression*. The expression editor lets you select a common characteristic; you can then add individual objects to the selection or delete them. You can even specify several characteristics in combination.

Communication

QGIS supports various linking routines and also provides the ability to import and export geodata. Since geodata usually exists in tables or databases, you can also create individual links to an external datasets. To save a layer in a database for export, you can create a file in GeoPackage format.

To create a new GeoPackage, click on *New GeoPackage Layer…* in the upper-left corner of the toolbar or select the function of the same name from the *Layer* | *Create Layer* menu. Enter the name of the database, the data table, the field name, and the type and length of the field. QGIS then copies the individual fields into the attribute table in the list area at the bottom of the window. After clicking *OK*, the software creates the GeoPackage and a corresponding layer in the layer view.

Conclusions

QGIS is the killer solution for viewing, creating, and visualizing geographic data. The software provides an enormous range of functions. A logical and intuitive user interface makes QGIS an easy option for hobby geographers as well as professionals. ■■■

Info

[1] QGIS: *https://www.qgis.org*

[2] Installation: *https://www.qgis.org/en/site/forusers/alldownloads.html#linux*

[3] URL page for tile server: *https://wiki.openstreetmap.org/wiki/Tile_servers*

[4] Shapefile format: *https://en.wikipedia.org/wiki/Shapefile*


OPEN SOURCE BACKUP
CONFERENCE
SEPTEMBER 26, 2018 | COLOGNE
GET UPDATED ON THE
LATEST DEVELOPMENTS IN
OS BACKUP
TECHNOLOGIES
REGISTER NOW!
OSBCONF.ORG
BAREOS | REAR | AMANDA

QMapShack visualizes maps and tracks

Map Viewer

QMapShack displays a wide variety of maps, making it easy to plan and manage your journeys. *By Karsten Günther*

QLandkart GT was once considered the best tool for displaying free maps on Linux and loading them on Garmin devices. Map enthusiasts were quite disappointed when the project came to a halt a few years ago, but fortunately, a number of developers quickly came together and launched a new project that extended the functions of QLandkarte.

QMapShack [1] lets you display map data from various sources, both online and offline. You can visualize, create, and edit GPX tracks and routes. Automatic routing is even possible. QMapShack combines information from different sources as a "project," which makes the program interesting for larger projects with several different elements. All in all, QMapShack is best described as a universal map planning and routing tool.

QMapShack Hands On

Some Linux package repositories contain packages for QMapShack, but they might not have the latest version. See the box entitled "Building QMapShack" for a brief look at how to build a QMapShack application yourself.

QMapShack comes up with an intuitive user interface (Figure 1). The most important functions are in the menu and toolbar. QMapShack supports various online and offline map formats, especially VRT, TMS, WMTS, RMAP, and Garmin IMG (see the "GMap" box).

Configuration

The setup process for QMapShack has improved considerably since the last version. In the past, you had to create special directories for many aspects of the program and

Listing 1: Installing QMapShack

```
$ hg clone https://bitbucket.org/maproom/qmapshack QMapShack
$ mkdir build_QMapShack
$ cd build_QMapShack
$ ccmake ../QMapShack
$ make
$ sudo make install
```

Building QMapShack

Most current distributions do not yet include the current QMapShack version 1.11.0 in the package sources, so you usually have to build the latest version of the application yourself. You need the Mercurial versioning system (version 4, or preferably 4.5) and CMake or CCMake v3.1. You also need GDAL, PROJ.4, and Routino, as well as QuaZIP and Qt v5.

Listing 1 shows the individual steps. First, clone the Mercurial repository to the local disk. Then create a working directory and change to the new folder. Now use CCMake to create a makefile: Create the configuration in the Ncurses interface by pressing *C*. Press *G* to create the makefiles and *Q* to end CCMake.

The `make` call builds the programs under `bin/`. This step creates several binaries: the main program `qmapshack`, the `qmaptool` utility for displaying maps, and the `qmt_map2jnx` and `qmt_rgb3pct` tools for converting data formats. `sudo make install` lets you import the program to `/usr/local/` per the previously defined settings.

If necessary, you can update QMapShack directly from the cloned directory. Change back to the working directory and get the current code by typing:

```
hg pull && hg update
```

Re-running `make` generates the new version, and `sudo make install` installs the update.

enter them in the configuration. The current version simplifies this step: Using the *Give me a path…* link on the welcome page (Figure 1), you just enter a base directory, and all the required folders are automatically created by QMapShack and entered in the configuration. Keep in mind that 10GB or more can quickly accumulate below the base directory. You may therefore need to distribute the different directories across several drives.

Offline maps are configured using *File | Setup map paths* (Figure 2). When you get there, you will see the path under which the software saves the map tiles (in the standard configuration, the default is `~/.QMapShack/`). The application only loads maps from the specified folder, not from its subfolders.

If you do not have offline maps, you can also revert to online maps. For example, `*.tms` ("Tile Map Server") files are used with online map services. The button with the plus symbol lets you add maps and map directories. When you create new `.tms` files, they will appear in the left view window of QMapShack under the file name you specify. The program won't distinguish identical file

GMap

One of the special features of QMapShack is the support for Garmin IMG files. This features is limited to maps created by `mkgmap` for Garmin devices, but QMapShack handles them very well. Testing an IMG file with QMapShack works much better than trying to test it with a navigation device. In addition, you can use the IMG info offline with QMapShack (i.e., exactly where it is important: outdoors, where no Internet is available). Currently the developers are working on the possibility of integrating navigation devices, such as GPS coordinate generators, with QMapShack.

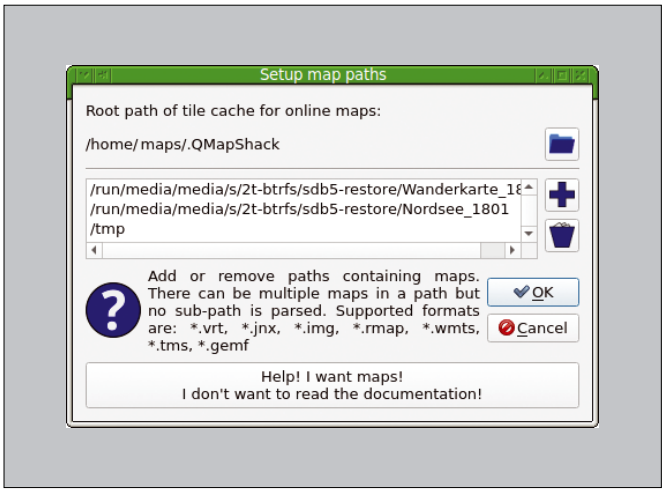


Figure 2: Before using QMapShack, you'll need to complete the configuration: First enter one or more directories, in which you will be saving map material.

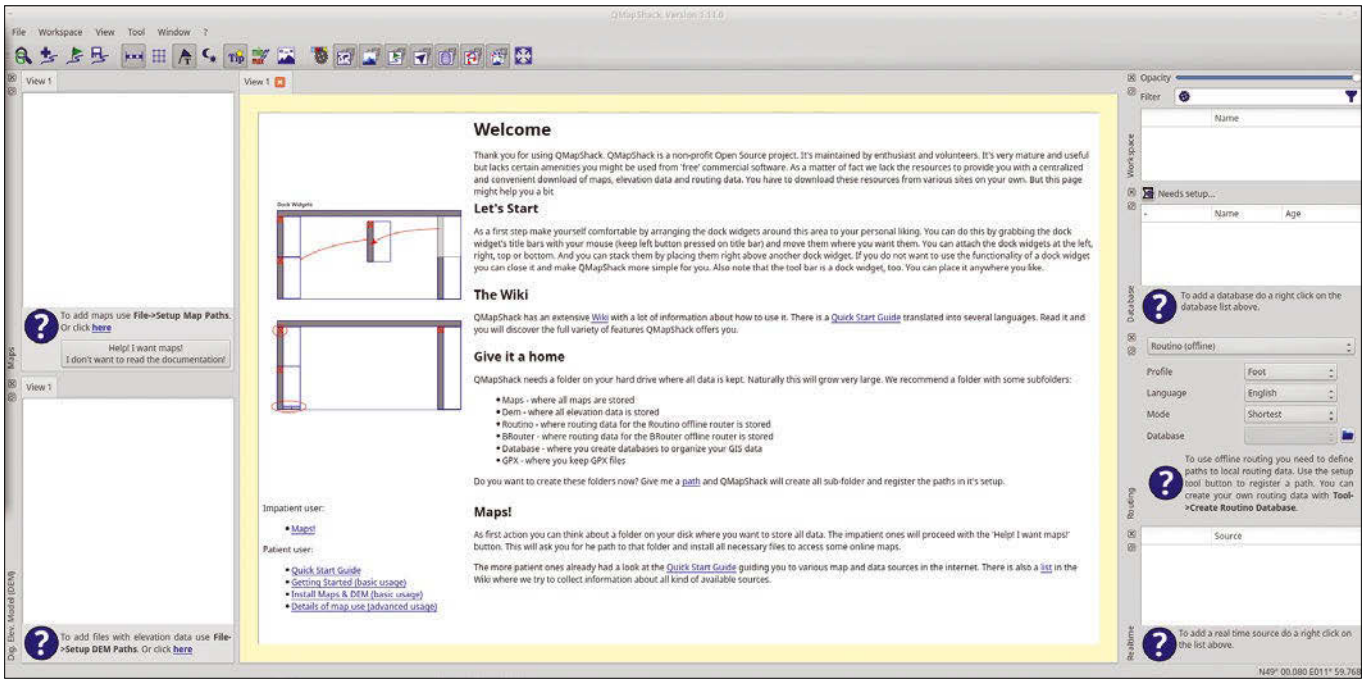


Figure 1: Views and settings structure the QMapShack interface. The current version includes an easy-to-understand manual for setting up the program.

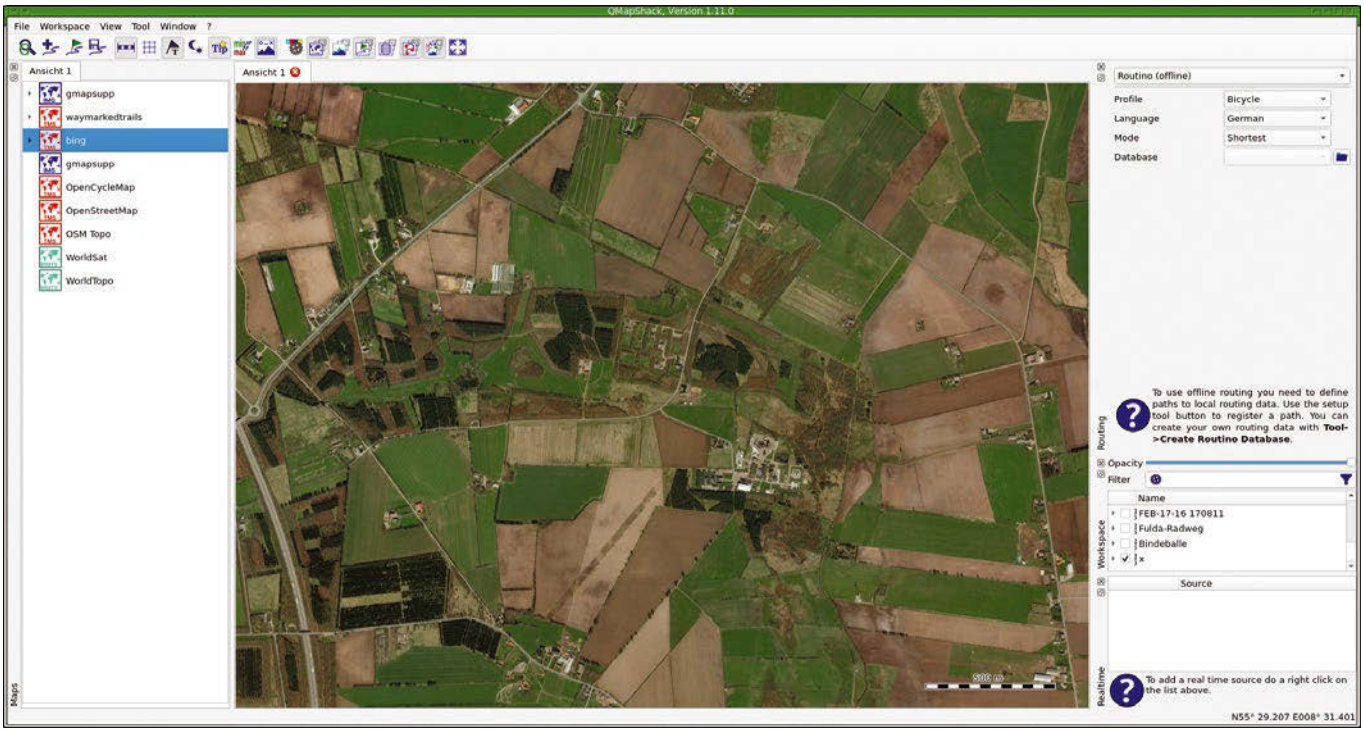


Figure 3: QMapShack can integrate several maps at the same time and display them one on top of the other.

names in different directories, so make sure you don't give two files the same name (Figure 3).

In addition to real maps, QMapShack also displays arbitrary tiles (e.g., the aerial images provided by Microsoft on Bing).

Listing 2: Integrating Tiles

```
<TMS>
<Title>Bing</Title>
<Layer idx="0">
<Script><![CDATA[
(
function convert(z1, x1, y1) {
  serverpart = 0
  serverpart = (serverpart + 1) % 4;
  function encodeQuadTree(zoom, tilex, tiley) {
    var tileNum = []
    for (var i = zoom - 1; i >= 0; i--) {
      var num = (tilex % 2) | ((tiley % 2) << 1);
      tileNum[i] = new String(num);
      tilex >>= 1;
      tiley >>= 1;
    }
    return tileNum.join("");
  }
  return "http://ecn.t" + serverpart +
    ".tiles.virtualearth.net/tiles/a" +
    encodeQuadTree(z1,x1,y1) + ".jpeg?g=1036";
}
)
]]></Script>
</Layer>
<Copyright>Microsoft - Bing</Copyright>
</TMS>
```

Integrating these images is not exactly intuitive, as Listing 2 shows. Please note that the legal situation is not clear for all map sources: Google does not allow the use of its tiles with programs other than Google Earth or its own online services. In addition, some of the maps are not very practical and only cover small areas or provide small zoom ranges. Others, such as the Alpine map, have a considerable scope and show many details.

In addition to pure map data, QMapShack also supports the use of Digital Elevation Model (DEM) data, which refers to digital elevation or terrain data. A whole series of servers provide DEM data that you can integrate into QMapShack as online maps. The original documentation dedicates a separate chapter [2] to the topic of DEM maps.

API Key for OpenCycleMap

The OpenCycleMap server operated by *Thunderforest.com* (and some others) requires an API key to deliver tiles without an *API Key Required* message (Figure 4). You receive the key free of charge [6] after registration as part of a noncommercial project. After a confirmation email, you will find the API key in the service's dashboard.

You then need to configure QMapShack so that the application sends the API key to the OpenCycleMap tile server with every request. Open the `OpenCycleMap.tms` file in the map directory, which only contains a few lines of XML code (Listing 3). In the `<ServerUrl>` line, add the API key with `?apikey=API key` before the closing `</ServerUrl>`. This modification gives you "undisturbed" tiles from the OpenCycleMap server again.

Where the `OpenCycleMap.tms` file is located in the configuration files depends on several factors and can be specified with `Setup map paths` during configuration. When importing old data from QLandkarte, you will often find it below `~/qlandkartegt-Konto/OpenCycleMap.tms`. If necessary, search for this file using `find ~ -iname OpenCycleMap.tms`.

Listing 3: Setting Up OpenCycleMap

```
<TMS>
<Title>Opencyclemap</Title>
<MinZoomLevel>1</MinZoomLevel>
<MaxZoomLevel>1024</MaxZoomLevel>
<Layer idx="0">
   <ServerUrl>http://a.tile.thunderforest.com/cycle/%1/%2/%3.png?apikey=<API_key></ServerUrl>
</Layer>
<Copyright>Map data: (c) OpenStreetMap contributors, ODbL | Rendering: (c) OpenCycleMap , CC-BY-SA</Copyright>
</TMS>
```

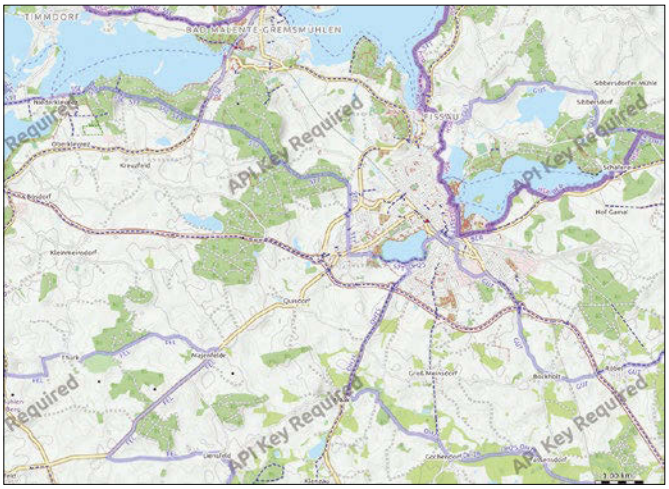


Figure 4: A number of servers mark the maps with a watermark if you do not specify an API key.

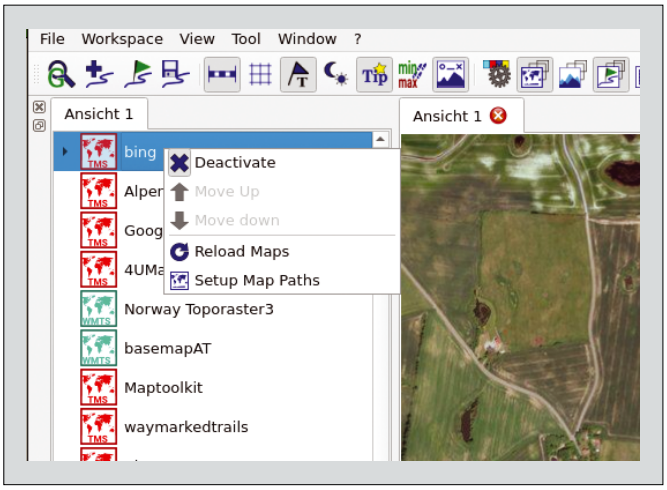


Figure 5: You can enable and disable maps via the context menu.

Navigating in the map window is quite intuitive: Hold down the middle mouse button to move the displayed map section, and use the mouse wheel to scale the view size. (By the way, you can use the same feature when creating tracks and routes.)

Services such as OpenCycleMap place a watermark on tiles until the user registers and obtains an API key (see the box entitled "API Key for OpenCycleMap.")

Installing Maps

Manually installed maps must first be activated. To activate or deactivate a map, right-click on the map in the list and choose the function in the context menu (Figure 5). QMapShack always sorts enabled map views at the top of the list.

QMapShack lets you enable multiple maps at the same time. For example, you could superimpose an aerial image on a map display or vice versa. This feature allows completely new forms of "realistic" map images, which, until now, could only be created by combining the maps with image-processing techniques.

The order of the maps plays an important role: Maps further down the list cover the maps above. The opacity of each activated map is controlled by a hidden dialog that opens after a mouse click on the small triangle in front of the map icon (Figure 6). The uppermost slider controls the opacity of the layer.

You also need to pay attention to the two sliders below: *Cache Size (MB)* limits the amount of disk space used per map view, and *Expiration (Days)* automatically removes obsolete tiles and replaces them later when the region is called.

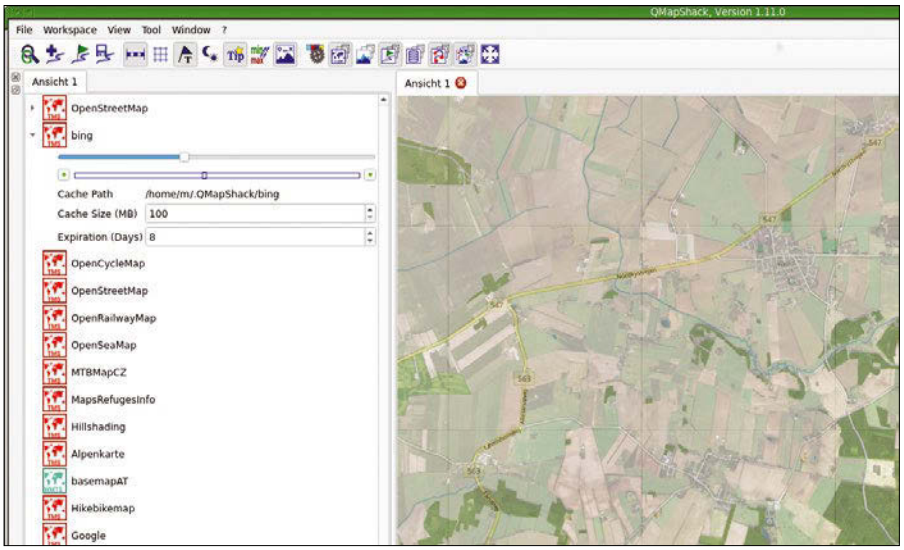


Figure 6: You can adjust the opacity of map displays to allow partial overlapping of maps and aerial photos.

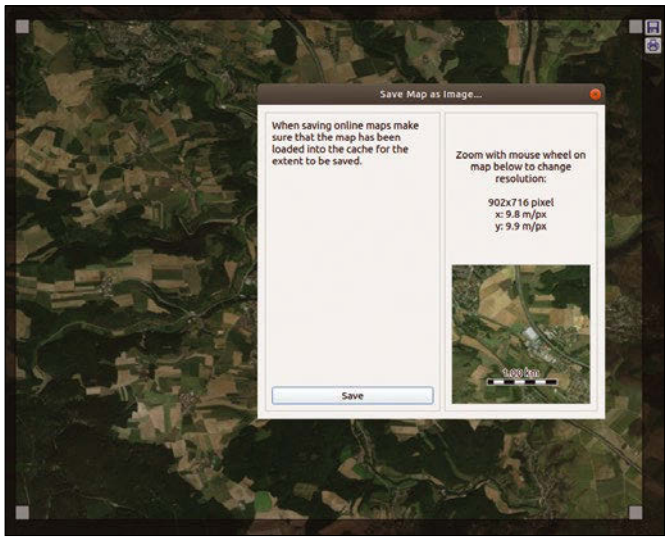


Figure 7: Using the print function, you can put the map on paper and also save the desired section as an image.

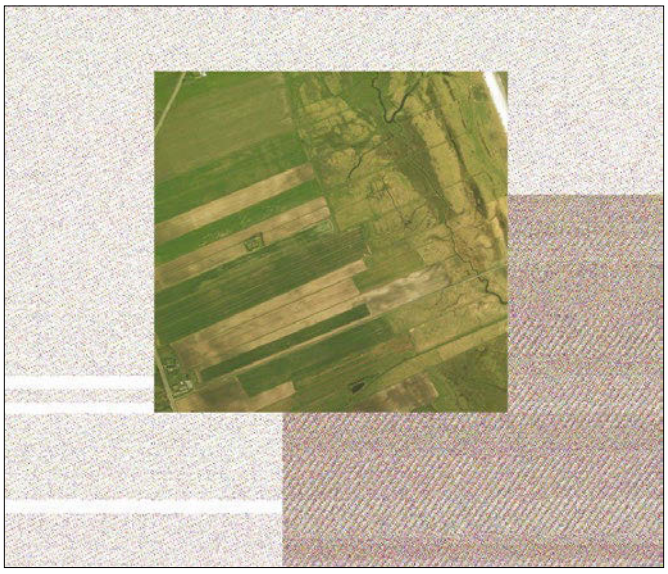


Figure 8: QMapShack displays OpenStreetMap and OpenCycleMap results correctly; however, errors sometimes appear in the image if you try to superimpose maps.

Print and Save

You can use Ctrl + P or the *File* | *Print* menu item to activate the print and save function for the displayed map section (Figure 7).

First select the desired map section and use the buttons at the top-right to start saving or printing. This works for the map representations themselves, as well as for aerial photographs and superimposed representations. However, the preview only shows the area you are really editing to a very limited extent. QMapShack shows the real resolution in meters per pixel. When saving, you will get a screenshot (also for aerial photos), but the quality is not convincing (Figure 8).

The actual printing function is even less satisfactory: Despite activated color printing and printing to file, QMapShack only delivered a black-and-white print with a wide margin in the test. In combination with OpenStreetMap and OpenCycleMap, however, both saving and printing worked well.

Tracks and Routes

One of the areas of application for QMapShack is convenient planning of routes on the screen. To create a track, right-click in the *Map* window and select *Add Track* from the context menu (Figure 9). This step activates "rubber band mode" and creates a targeted line, which you place on the map.

Right-click to exit this mode, and then save the created track to disk as a GPX track using *Save as New* (Figure 10). Saved tracks automatically appear in the project overview, which QMapShack displays by default at the bottom of the right sidebar.

You can also integrate portals like *Waymarkedtrails.org* into QMapShack. To do so, press Ctrl + L or select the *File* | *Load GIS data* menu item. Among other things, the application supports importing GPX and TCX files; it can also handle the new FIT format used by today's Garmin devices. The corresponding tracks also appear in the project overview. If the track disturbs the map view too much, use the *Transparency* slider to adjust the display.

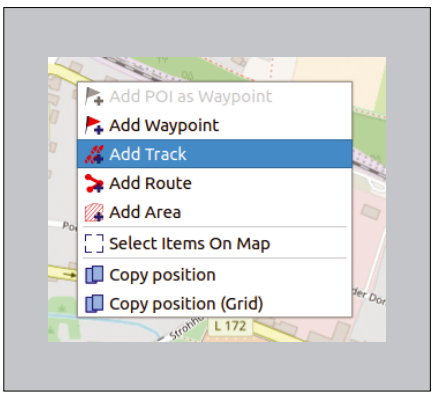


Figure 9: Right-click on the map to open a context menu, where you can create tracks and routes.

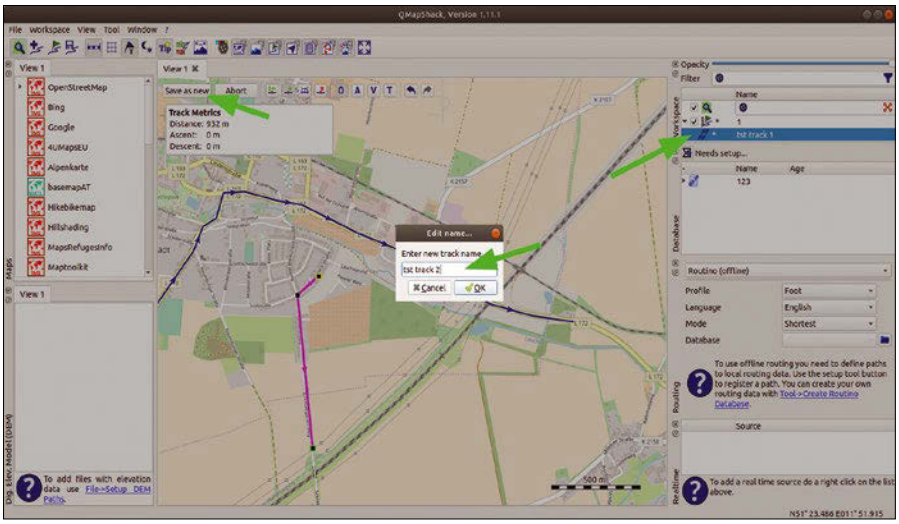


Figure 10: Click *Save as New* to save the track. The saved track will then appear in the project overview.

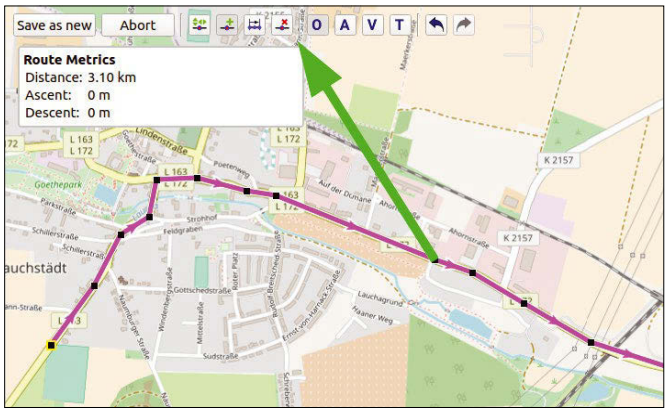


Figure 11: The route editor lets you adjust the waypoints of a previously defined route.

You create routes in the same way as tracks – via the context menu. Select the *Add route* option, put together the route, and save it. Unlike a track, a route can be corrected at any time.

Installing Routino

Routino is the (offline) router preferred by QMapShack. The program is available in the package sources of many distributions. In some environments, however, you may need to build the application manually in order to use the latest version. Listing 4 shows the process for Ubuntu.

You load the routing data from the Internet [7]. The corresponding archive contains four files with a total size of more than 5GB, which you unpack into the `qmapshack-data/Routino/` directory. Routino also supports PFB files, such as those offered by Geofabrik [8]. Activate routing by selecting the router, profile, and database (Figure 12).

Listing 4: Installing Routino

```
$ sudo apt install gcc make libc6-dev libz-dev libbz2-dev subversion
$ svn co http://routino.org/svn/trunk routino
$ cd routino && make && make install
```

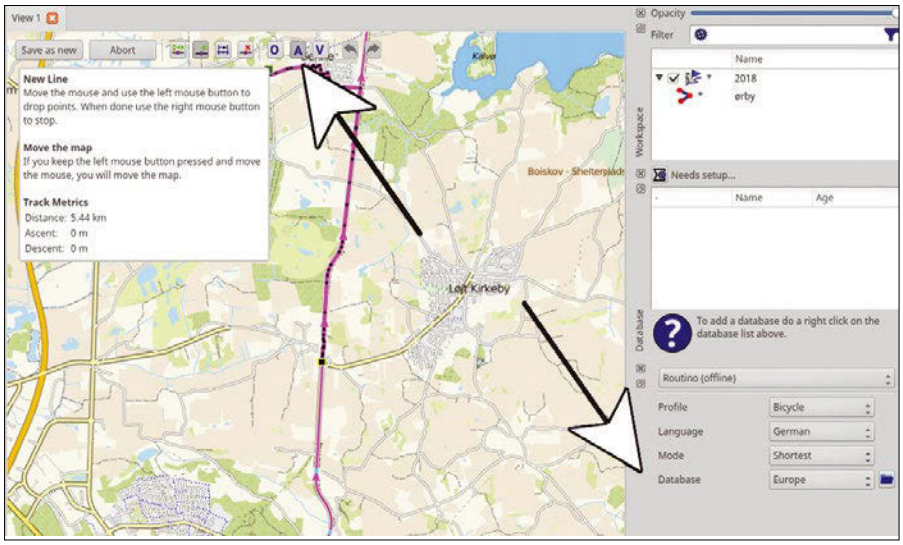


Figure 12: After you set up the database, Routino works automatically as soon as you create the router in *A* mode.

You can reactivate the editor mode by selecting the route with the left mouse button and clicking on the *Move route points* icon (Figure 11).

Routing with Routino

For additional options when planning routes, set up the Routino [3] open source router (see the "Installing Routino" box). The program calculates the connection between two points on the map based on topographical data. However, you need additional data to enable routing on the local computer. Your options will be similar to those provided by online services like Naviki [4] or openrouteservice [5].

First, configure Routino by specifying a directory with the routing data. If you are using a Garmin map, you will need to install additional files: Routino cannot use the routing data available in `Gmapsupp.img`. Store the data in the configuration directory under `Routino/`. This directory can also be on another data carrier.

To calculate a route, first set the starting point, then activate automatic routing with button *A*, and then, based on the settings in the router dialog, you will see either the shortest or fastest route to the current position of the mouse pointer. Routino offers only very limited settings – Naviki or openrouteservice usually deliver significantly better results.

Conclusions

QMapShack is an interesting and powerful program for outdoor and mapping enthusiasts. Small things like the storage and printing functions, which do not work well in all situations, cloud the overall impression but are basically only additional features at the end of the day. The display features for online and offline maps work very well, as do the functions for creating and managing tracks and routes. After a training phase, QMapShack is very useful for planning and documenting tours. The integration of Routino also works surprisingly well, even though the routing function requires a large amount of additional storage space. ■■■

Info

[1] QMapShack: *https://bitbucket.org/maproom/qmapshack/overview*

[2] DEM maps: *https://bitbucket.org/maproom/qmapshack/wiki/DocBasicsMapDem*

[3] Routino: *http://routino.org*

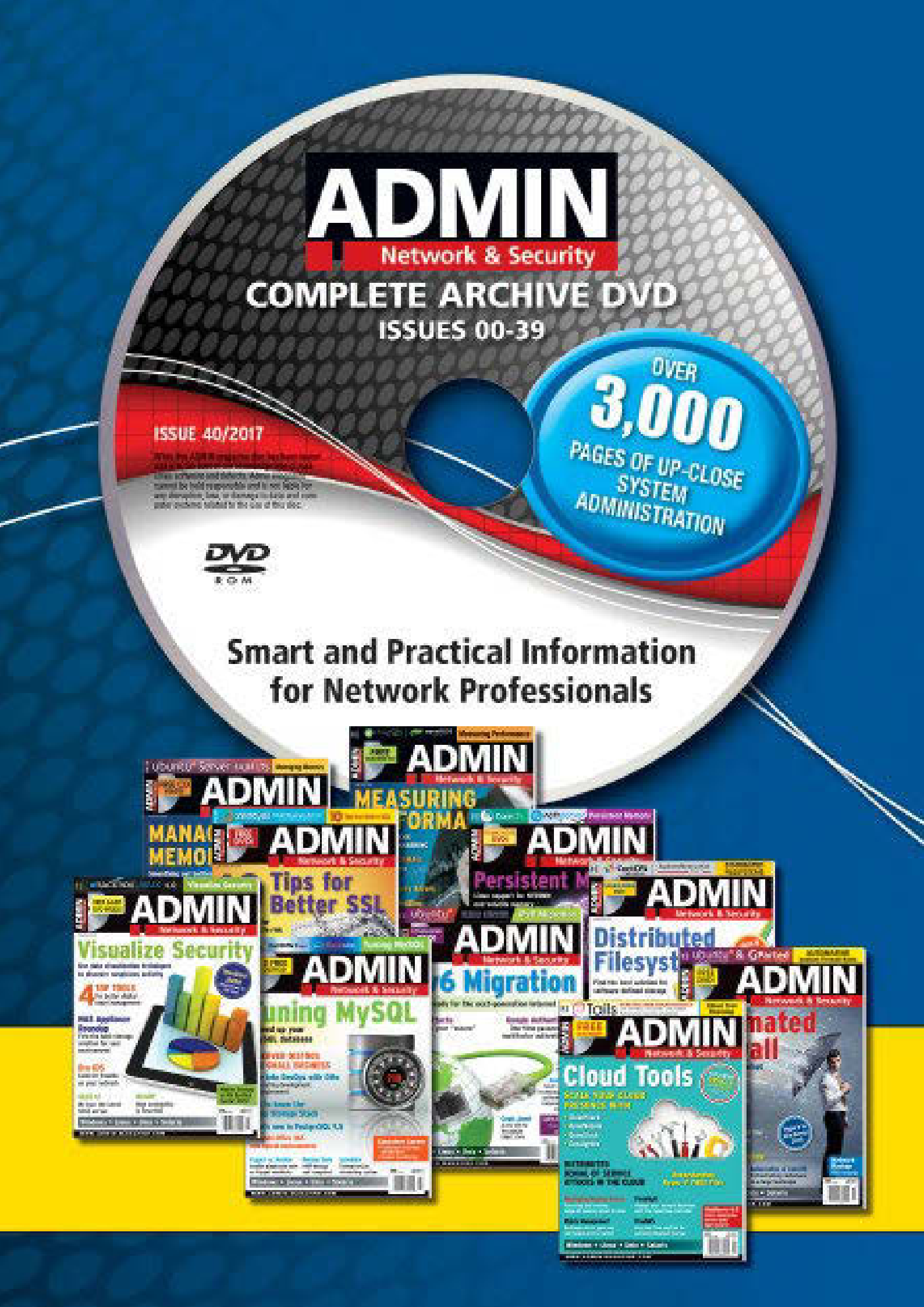
[4] Naviki: *https://www.naviki.org*

[5] openrouteservice: *https://openrouteservice.org/*

[6] API key request: *https://manage.thunderforest.com/users/sign_in*

[7] Routing data for Europe: *https://jeepxj.de/pub/routino/Europe.zip*

[8] Geofabrik PFB files: *http://download.geofabrik.de*


ADMIN
Network & Security
COMPLETE ARCHIVE DVD
ISSUES 00-39
OVER
3,000
PAGES OF UP-CLOSE SYSTEM ADMINISTRATION
ISSUE 40/2017
DVD
ROM
Smart and Practical Information for Network Professionals
Visualize Security
Tips for Better SSL
Tuning MySQL
Cloud Tools
Distributed Filesyst

Exploring Kubuntu 18.04 LTS and the Plasma desktop

Kubuntu Close-up

Kubuntu 18.04, which is built on Ubuntu Bionic Beaver, is the first long-term support release that includes the Plasma 5 desktop. This article looks at the latest Kubuntu and highlights some cool configuration tricks. *By Rick Timmis*

In July 2014, KDE announced Plasma 5.0, a visually updated core desktop experience that was user friendly and familiar. Plasma also introduced a new major version of KDE's workspace.

The new "Breeze" artwork concept introduced cleaner visuals and improved readability. Central workflows were streamlined, while well-known overarching interaction patterns were left intact. Plasma 5 improved support for high-DPI displays and added a "converged shell," able to switch between user experiences for different target devices. Changes under the hood included the migration to a new, fully hardware-accelerated graphics stack centered around an OpenGL ES scene graph. Plasma is built using the new Qt 5 and Frameworks 5.

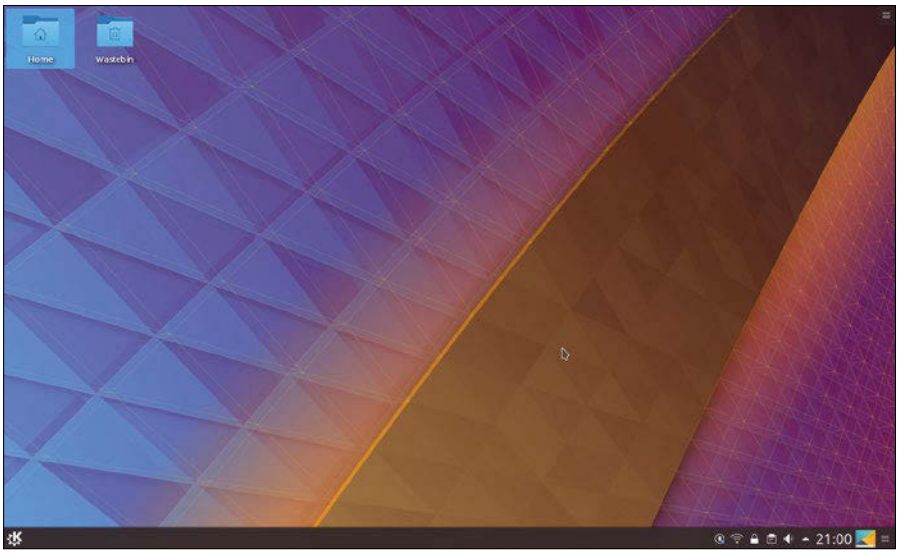


Figure 1: The default Kubuntu desktop.

The changes ushered in with Plasma 5 have finally reached the Kubuntu LTS series with Kubuntu 18.04. If you're new to Kubuntu, or even if you're upgrading now from a previous LTS edition, this article will give you a brief introduction to Kubuntu 18.04 and show you some highlights of the Plasma 5 desktop.

Transitions Take Time

Long-term Linux desktop users will remember KDE 3 and 4. These major platform transitions brought stunning improvements to the KDE framework and the underlying Qt (say "Cute") libraries. These transitions were an enormous

Figure 2: CPU Load monitoring widget, with its configuration menu activated.

undertaking, which was particularly burdensome for KDE – not only was the core desktop ported to the new Qt toolset, but it also required KDE Frameworks and the full set of applications. KDE applications represent a considerable portfolio of software, all of which must be ported to the new frameworks at each major release transition. Of course, users found such transitions frustrating, which resulted in a drop in the user base. However, with each transition, the KDE community delivered improvements that helped with recovering the lost user base.

Plasma Active Desktop

The Plasma desktop in Kubuntu blends together the dark and light breeze themes to create a beautiful and modern-looking desktop (Figure 1). Plasma uses vector-scaling graphics throughout, which provides smooth scaling on all displays. The ethos behind Plasma is one of an active desktop. *Active* means that the desktop should both embrace and assimilate your day-to-day workflow. This active desktop is achieved through the use of widgets (aka plasmoids) – small configurable desktop applets that offer a huge range of functions.

One of my favorite uses for Plasma widgets is to create specialized active desktops that are configured to a specific workflow. Recently, I have been working on the #100DaysOfCode challenge. To support this activity, I configured a selection of widgets on my desktop consisting of the Notes, Timer, and Browser widgets. These tools enable me to have the PHP documentation sitting directly on the desktop, alongside my notes and a countdown timer, which enables me to configure specific blocks of time to work on coding.

To set this up for your environment, begin by right clicking on the Plasma desktop and choose *+Add Widgets*. A panel on the left will slide out. The *Add Widgets* panel contains the default set of widgets shipped with Kubuntu. Deploying these widgets onto your desktop is as simple as dragging and dropping. Additionally, a vast array of widgets are available through the *Get New Widgets* button at the bottom of the panel.

Each widget carries its own specialized configuration, resize, and positioning functions. To access the configuration panel, hold down a long left click on the widget (Figure 2). The widgets scale smoothly. You can drag them into different locations across the desktop and organize them neatly to match your specialized workflow.

Plasma Activities

Activities are an often undiscovered and underutilized Kubuntu feature. Activities allow you to set up desktops that are specifically tailored to your work environment. You can think of activities as something like virtual machines or Docker images for your desktop. Activities have independent widget sets, wallpapers, and application separation. You can configure activities for software development, blogging, photography, and gaming. Access each of these activities with a simple click to switch.

As an example: Say you're working on a new KDE application using Qt Designer, and you've just kicked off a large compilation. Instead of opening another application to work on your blog, simply switch to your blogging activity. All your work continues running in the software development activity, leaving you with a clutter-free desktop to work on your blog. This feature becomes even more useful when you create multiple activities and configure them to match each of your workflows.

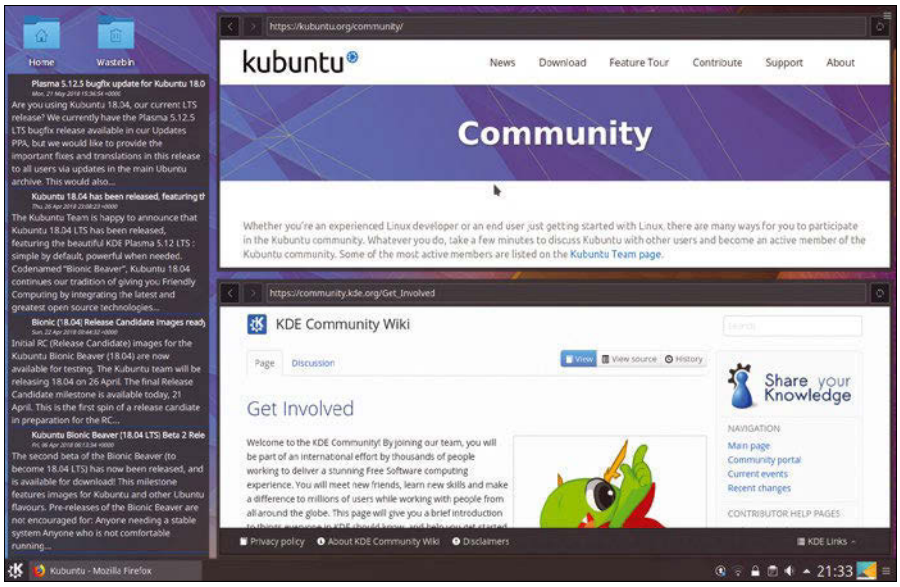


Figure 3: The configured Kubuntu Community activity.

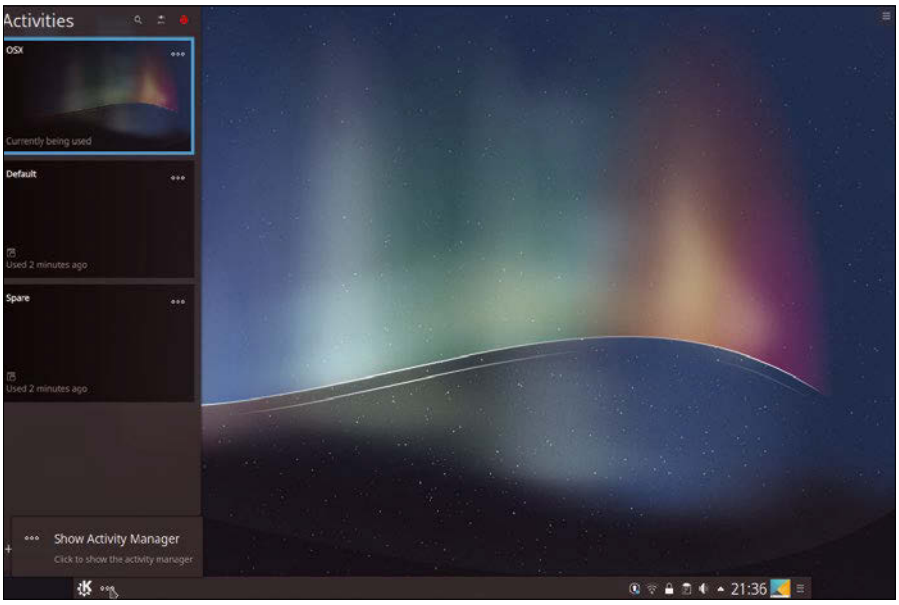


Figure 4: The OSX-inspired menu panel with an activity switcher.

Suppose you would like to create a Kubuntu Community activity and add some useful widgets that are connected to community resources. Just right-click on your desktop and click *Activities* from the menu. From the *Activities* panel (which slides out from the left), click on *Create Activity*. Name the activity *Kubuntu Community*, and click *Create*. Your new activity appears in the activity panel. Click on the activity to switch to it.

Once you are in your shiny new Kubuntu Community activity (Figure 3), you can configure the desktop, wallpaper, and widgets, and these settings will remain with this activity.

Now you can add some suitable widgets that give you access to the Kubuntu community [1]. Use the *Get New Widgets* button to install the RSS widget by MirceaKitsune. Then add 2 browser widgets and one RSS feed widget to your activity. Use each widgets' configuration to set the RSS feed URL to *http://kubuntu.org/feed/* and the browser widget URLs to *https://kubuntu.org/community/* and *https://community.kde.org/*

Personalizing Your Plasma Desktop

Kubuntu is all about being configurable, providing you with a wide range

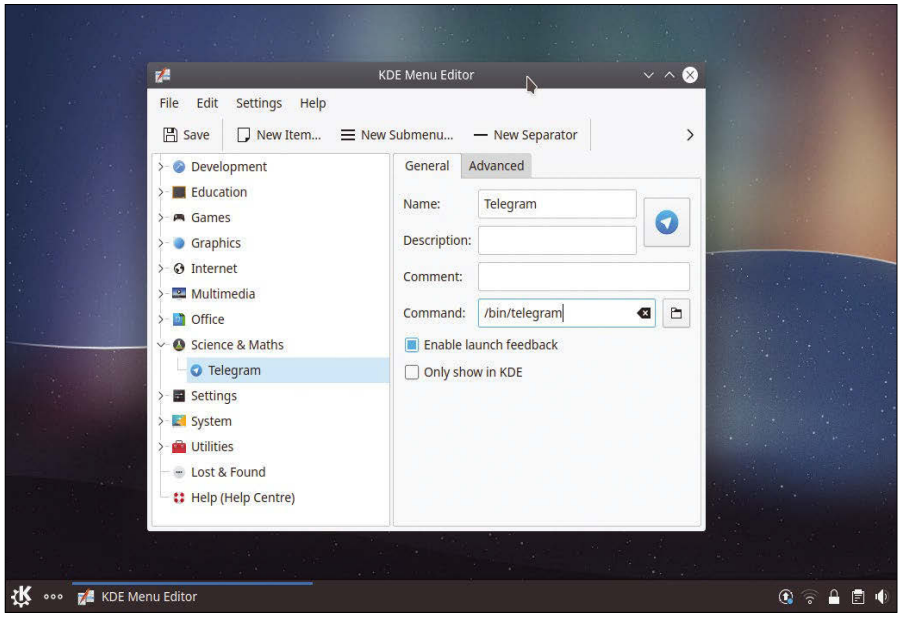


Figure 5: Configuring additional menu entries using KMenuEdit.

of configuration settings. With these options, you can configure almost any desktop paradigm.

By default, Kubuntu comes with three different Kickoff menus (the Plasma icon on the bottom left that launches the application chooser) To see the different styles, right-click on the Kickoff icon. Click on *Alternatives*, and you will be presented with a switcher that contains three choices:

- Application Dashboard: Provides a full-screen application chooser.
- Application Launcher: Enabled by default.
- Application Menu: An option familiar to all KDE aficionados.

Next you can change the bottom panel, making it a little smaller, shorter, and centered. This option is useful for getting a little extra screen real estate. On the very right of the bottom panel, click the hamburger menu (three horizontal bars). This option opens the configuration settings. On the right, you'll notice two arrows (left and right). The arrows allow you to shrink or stretch the panel.

On the left, you'll notice a double-ended arrow, which allows you to move the positioning of the panel from left to right. Shrink the panel a little (to suit your taste) and then position it (central is nice and rather OSX-like).

You can also add an activities launcher to make it easier to switch activities without having to right-click on the desktop.

From the panel configuration menu, click on *+Add Widgets*, and click and drag the *Activities* widget (which shows the activity manager) down onto the bottom panel. You'll notice that the icons slide around as you choose a position. Once you are happy with the location, left-click to drop the widget onto the panel. Voila! Now you have an activity switcher on your panel (Figure 4).

You can add many more widgets to the bottom panel using the same technique.

A Few Final Funky Features

So far I have only scratched the surface of what is possible with Kubuntu 18.04. I'll finish this article by presenting some really funky new features.

When you first install Kubuntu it runs a background process called Baloo. Baloo's job is to provide a Luciene index of your home directory. This index is used by the global menu to provide super-quick access to your files and directories.

To access the global menu, click Alt + Space, and the menu will appear top and center.

Type anything you like into the box, and the global menu will show you some intelligent results – whether it be launching an application, locating a file, or finding a contact. In fact, the global menu will even present you with potential applications that you might want to install from the KDE Discover Software Center. Now that's pretty funky wouldn't you agree?

Finally, I want to share a hidden gem with you. You might want to add extra items into your launcher menu. An example might be if you have downloaded an application from the Internet, and perhaps installed it yourself, and you find that there is no entry in the Application Launcher. Use the global menu Alt + Space and type *kmenuedit* into the search box. Click on *Run kmenuedit*. The KMenuEdit application (Figure 5) will let you add additional Application Launcher menu items with icons. ■■■

Author

Rick Timmis (*http://www.ricktimmis.com)* is a charismatic, optimistic, and sociable geek. He is an active participant in the free software and open source community, as well as a founding member and former CEO of the UK Open Source Consortium. He is currently a community manager, council member, and developer with the Kubuntu flavor of the Ubuntu linux distribution.

Info

[1] Kubuntu Community: *http://www.kubuntu.org/community*

Social networks in the enterprise

Inner Circle

If you want to use social media and microblogging platforms, you do not need to leave your data sovereignty to external corporations. This month, we look at free decentralized software alternatives to Facebook, Twitter, and other social media and examine whether they are suitable for use in the enterprise and other organizations. *By Erik Bärwaldt*

Social media and microblogging platforms can drive involvement and discussions between employees of a company and between employees and consumers. However, the suitability of well-known services such as Twitter or Facebook is limited. In addition to security concerns, data protection and user privacy are concerns, especially if the operators of the services then market their customers' data commercially.

Today, however, you can find services and software that promise improved security and – as open source projects – more transparency. In this article, I take a look at Diaspora [1], Friendica [2], GNU social [3], and Mastodon [4]. In addition to their general capabilities, I also examine the conditions for the internal use of these services in companies and organizations.

Criteria

The requirements for a microblogging service in the enterprise differ significantly from those for purely private use. For example, certain encryption methods are unavoidable. Advertising, tracking, and other unwanted content have no place in microblogging at work. Cross-platform availability is also important for users. Thus, the short messages should end up not only on the smartphone, but also on the workplace computer, and the computer must be able not only to receive tweets and posts but to send them.

Ideally, employees will not need to install an additional application to use such a service. In certain areas of the company, the possibility of sending or receiving multimedia content, such as individual pictures or short videos, is also relevant, in addition to pure text messages on a small scale.

However, very large companies and organizations often still rely on traditional proprietary solutions like IBM Lotus Quickr, Lotus Connections, Salesforce, or Microsoft SharePoint Services. Instances of these products are usually operated in the corporate data center; they are complex to manage and, what's more, functionally often significantly oversized – at least for small to medium-sized companies and organizations. For the latter, heavyweights like these are usually not suitable for cost reasons alone.

Diaspora

Diaspora [1], written in the Ruby programming language and published under the AGPL license, is one of the more popular social networks with functionality similar to Facebook. Unlike Facebook or Twitter, Diaspora is not a centralized service. Instead, it comprises many decentralized servers (known as pods). The software offers possibilities for both time-staggered and real-time communication.

Thanks to a modular and decentralized structure, the admin can extend the application's functionally and retain control over the data distributed by it.

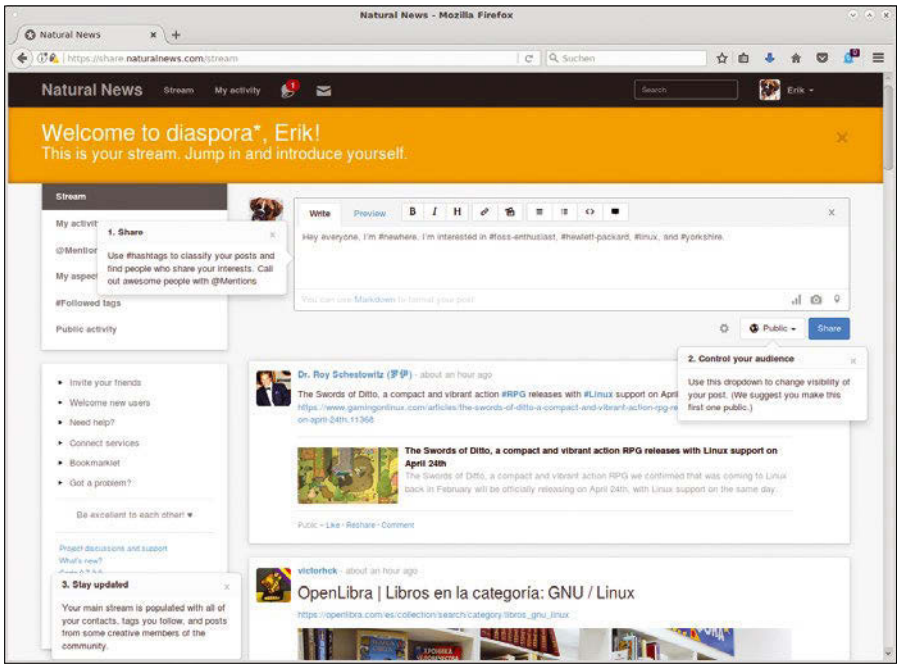


Figure 1: The Diaspora start page seems a bit confusing at first.

Online

A list of existing Diaspora servers (pods) can be displayed by clicking the *Join us!* button on the web page and then selecting the *Sign up* option. If you click on the corresponding URL, you end up on the start page of the Diaspora server. When you get there, create an account, for which you only need an email address, a nickname, and a password. Once you have logged in with this data, you can upload a photo to the profile page and enter various tags, which roughly describe your areas of interest, through which others can find you.

The main Diaspora page initially seems a bit overloaded with streams, but it ultimately follows a familiar structure. The dashboard is arranged vertically on the left. To the right is the message area. At the top, a horizontal action bar has a drop-down menu for the user's settings options. You can reply to messages immediately without having to open a separate input area (Figure 1).

Profile

You can change your profile settings in Diaspora to suit your taste by clicking on your name in the upper right corner of the browser window and selecting the appropriate options in the menu. Diaspora offers various switches concerning privacy and security. However, some of these are somewhat illogically arranged in the menus, and you will sometimes have to search for them.

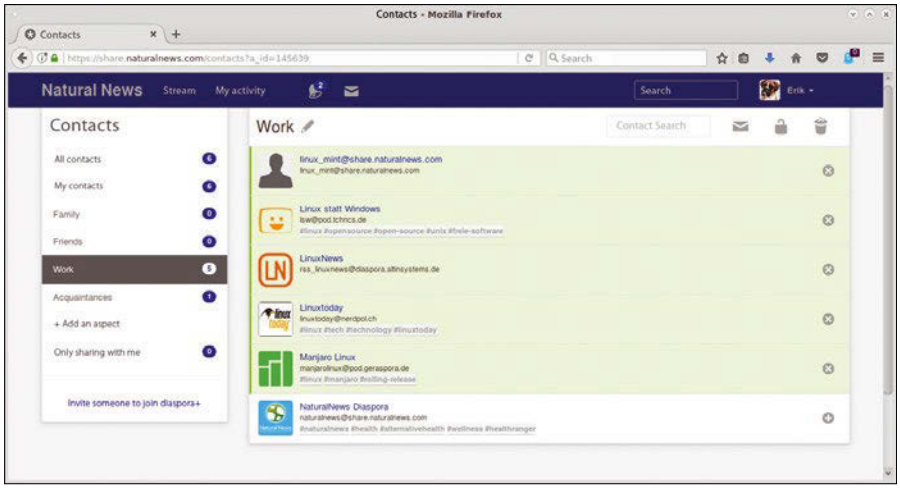


Figure 2: Diaspora makes managing contacts very easy.

Among other things, the software offers the option to remove metadata from uploaded photos that would allow the user to be identified. You can also enable the NSFW option, which often refers to pornographic content that is "not suitable for the workplace."

Very detailed settings are available for email notifications – all enabled by default. If you use the network intensively, this can lead to a large amount of received email. It makes sense to limit the settings so that Diaspora only sends really important notifications.

Contacts

Diaspora offers a very simple option for adding new contacts to your profile with the *Contacts* entry in the user menu. You can call up a list of all participants who share your interests by keyword and then add the desired contacts from the list by clicking on *Add contact* to add it to a group of communication partners. By default, Diaspora already has several groups, which it lists vertically in the dashboard on the left. The number of contacts is listed next to each one.

If you want to delete contacts from this list, you can do so with the use of groups. If you click on a group, its members appear on the right side of the window. Behind each contact is an *x* button to remove it. However, it does not disappear immediately: only after you open the list again (Figure 2).

Server Installation

Diaspora can also be installed on a dedicated server, which turns it into a pod for public use. The developers do not envisage isolated in-house use (e.g., for a department or branch in a company) [5]. Nevertheless, the admin and the users of a pod can set up private groups that are not publicly accessible through profile settings.

The specified hardware requirements are 1.5GB of RAM, swap space of 1GB, and a multicore CPU. Sufficient mass storage must also be available, especially with frequent uploads of multimedia content.

The Diaspora website provides detailed information [6] on server installation for various Linux distributions. Because setting up your own pod is very complex because of the numerous requirements, an installation package

is under development. However, it had not yet been released for production at the time this article was written. As far as possible, the developers try to support problems on their Freenode IRC channel.

Friendica

Friendica [2] is the all-rounder among microblogging platforms. Like other freely licensed microblogging platforms, the AGPL software, continuously developed since 2010, is organized decentrally in instances that each run on their own servers. Friendica has a modular structure, to which the user adds functions via plugins.

Of all the features, the good interconnectivity of the service stands out: Friendica also supports GNU social, Diaspora, Twitter, and Pump.io accounts [7]. A user can operate several profiles at Friendica and – if they register them under the same email address – switch between them. Numerous other platforms, on the other hand, commit users to their own offerings. Only independently developed clients like Choqok [8] allow you to use these platforms with external software and create accounts for them.

Friendica also lets you to post multimedia content. As a rule, the web browser is used as the user interface, but third-party clients are also increasingly integrating with Friendica (e.g., Choqok since version 1.6). Apps for mobile devices are also available.

Getting Started

If you want to become active at Friendica, you can – but don't have to – set up your own server. The project's website maintains a detailed list of decentralized public access nodes [9] offering various features. Users simply log on to one of the servers from a web browser. You can do this on the respective start pages using the *Sign up now* or *Create account* buttons in the browser window.

Friendica requires only a full name (even fictitious ones are allowed), a nickname, and the user's email address to register. If a user profile already exists on another Friendica server, the user can also import this. After clicking on *Sign up now*, the service automatically sends email with the newly generated password to the user. With this and your username, you can then log on.

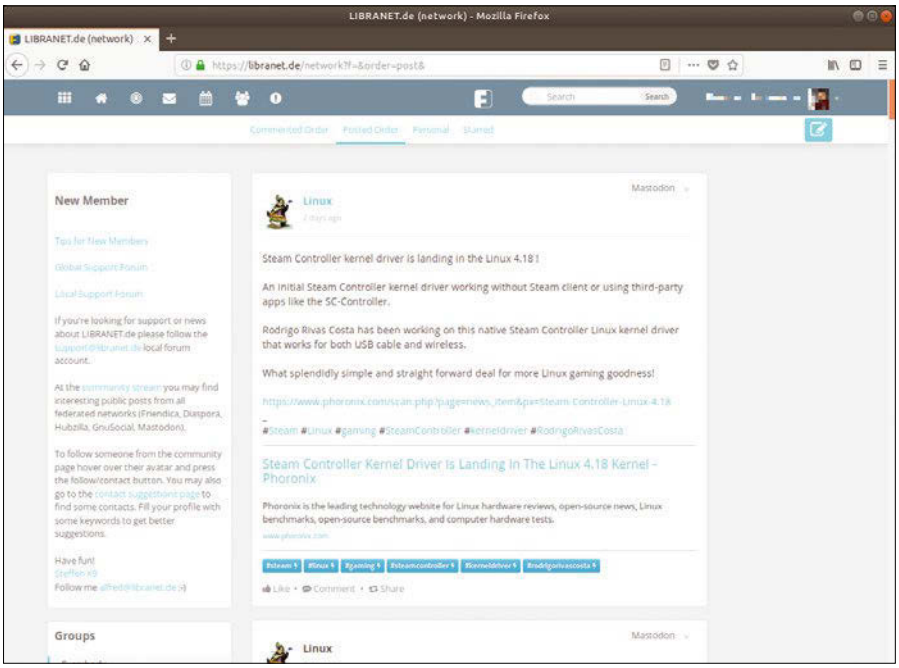


Figure 3: Friendica's main window appears clearly structured.

Profiles

Configuring the basic settings for your profile involves adding an avatar and optionally entering an address and keywords. These are keywords with which a user outlines their interests and hobbies. If you then search for one or more keywords on the platform, you should be able to find the respective user. The dialog also lets you hide contacts and friends. In this way, other users do not see with whom a participant communicates.

Once you have set up and saved the profile, you should click on the username in the upper right corner of the window and select the *Settings* option in the context menu. Then replace the password assigned by the system with a personal password. In *Basic Settings* you can localize your setup, so the menus and input fields appear in a language of your choice.

In the *Security and Privacy Settings* menu, you can enable or disable options. Do you want the profile to be visible to other users or on other servers in the network? Should other users be allowed to write on your bulletin

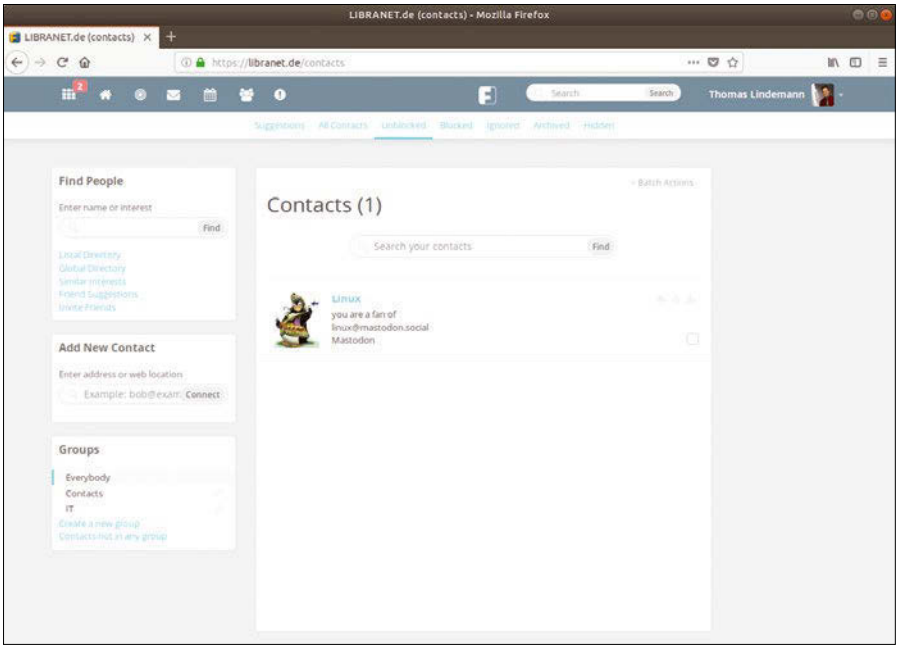


Figure 4: Friendica manages contacts in a uniform interface.

Figure 5: Hotot allows the simultaneous use of different services.

board? Should third parties be allowed to tag their own contributions? Some options for private messages and spam prevention can also be set. All configurations can be switched on and off by slide control.

Friendica also offers the option in the settings dialogs to define a default group membership for new contacts and configure notifications in a separate dialog (Figure 3).

Contacts

Friendica lets you integrate contacts from other microblogging services in your browser. In this way, you can subscribe to content from Instagram, Twitter, GNU social, Pump.io, Diaspora, and others with the traditional *Add New Contact*. It should be noted that although contacts can be loaded, this does not enable direct private communication, and the user must have their own account with the source servers.

Forwarding of public messages is also integrated, and users can receive automatic notifications of new messages for subscribed microblogs from external services outside of the Friendica network. After linking, all contacts appear in the Contacts area, regardless of the service used. You can also add RSS feeds, so Friendica also can replace a conventional RSS reader (Figure 4).

Groups

With the help of a group editor, which is accessed through the Contacts page on the left side of the dashboard, admins can add individual contacts to different groups. By default, all contacts are in the *Friends* group. You create groups by choosing *Create New Group* and then transfer the individual members to the new group by double-clicking; they remain in the standard group at the same time.

If a user only wants to view the feeds of a specific group, they can click on these feeds in the Network pane of the dashboard on the left. Where all new feeds were previously displayed, only the feeds from members of this group now appear.

Friendica lets you use and manage multiple profiles simultaneously, each with their own avatars. The operator can also show or hide the individual profiles.

Hotot [10] is also available as a desktop client for Friendica and is in the repositories of many distributions. Hotot can also be used with the Identi.ca service and Twitter; the client comes with a firewall and filter settings for unwanted content (Figure 5).

Installing the Server

On the Friendica site, you can download the source code for the server either as a `tar.gz` or `.zip` archive from GitHub by clicking on *Download*. The *Resources* | *Installation* item takes you to a short instruction on how to install the server. If the server is to be publicly accessible, you need to register a domain or a subdomain intended for this purpose in advance – as with all microblogging systems.

Friendica's software requirements also include a LAMP stack with an Apache web server, a MySQL database, and support for the PHP scripting language version 5.4 or newer. Alternatively, the Apache server can be replaced by Nginx [11] or Lighttpd [12]. Friendica does not force you to run MySQL as the database, either: You can use MariaDB.

Minimum hardware requirements are not specified by the developers because the platform also runs on virtual servers in the cloud. Friendica provides a quick guide to installing the server and the software requirements in the main window in the *Resources* menu. More detailed instructions can be found in the archives with the source code.

GNU Social

The GNU social project [3], previously known as StatusNet [13], is decentralized and seeks to escape the influence of governments and organizations. It has been around since 2010 and is licensed under the AGPL. GNU social is driven by independent servers known as nodes. A list of different groups with the corresponding URLs can be found online [14].

Individual members can post their text messages to multiple nodes on the network and belong to multiple groups. Although the software can be used across platforms in a web browser, it also has

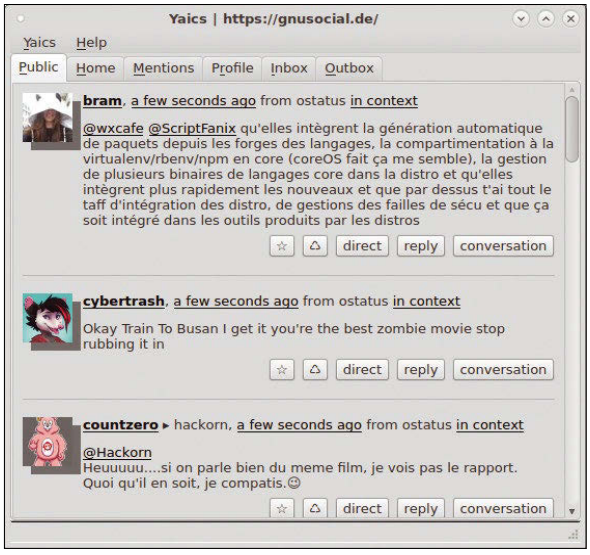


Figure 6: External clients are also available for GNU social; one of the newer access options is Yaics.

its own clients for different operating systems. Programmers develop and maintain them independent of the actual GNU social Project.

On Linux, the KDE application Choqok [8] and the Gtk+-based Heybuddy [15] program are particularly noteworthy, and Yaics [16] is a new development (Figure 6).

Group Dynamic

Users operate GNU social nodes with different interfaces, which are only marginally visually different. New members usually register and log in from the main window. In some cases, this is done with a dedicated *Login* button, which branches to a login and registration dialog.

To log in to a publicly accessible GNU social server, the user does not need to provide detailed personal information. All you have to provide is your email address and a username and password. The username must not contain any capital letters or special characters. You can then use the service immediately; verification of your email address with a confirmation message is not necessary.

You can compose individual short messages (quips) by clicking the pencil icon in the top right corner of the browser window. The messages can have a length of up to 1,024 characters. Direct input of a quip is also possible in an input box on the left side of the window. The individual quips that users compose appear one below the other in the browser window in the timeline and include photos. If you have joined individual groups, the timeline shows quips from the members of these groups. The display can be configured.

To make more detailed settings and to log out, press the *Profile* button in the title bar at the top of the browser window beside the free text search box.

Because GNU social is decentralized, users can only link tags to the server to which they send their quips. Conversely, contacts from other servers within the GNU social network also appear in a user's timeline. However, this requires a corresponding subscription on the source server.

GNU social offers the option to create your own groups, which the administrator and founder of the group can also mark as private. To do this, add a checkmark in front of the *Private* option in the settings dialog. The administrator is also responsible for adding new group members. With the help of two further options, the admin also defines how a group handles the sending and receiving of private messages. Several selection options are available (Figure 7).

Own Server

GNU social is under a free license, so companies and organizations that do not want to communicate in public space through third-party servers can integrate the service into their own intranet. To do this, however, they have to fulfil numerous conditions: Besides PHP scripting language version 5.5 or newer, MariaDB version 5 or newer and a web server are required.

The web server can be Apache, Nginx, or Lighttpd and must support HTTPS connections, which additionally require a signed certificate (free of charge, e.g., from the Let's Encrypt project [17]).

For the PHP scripting language the following extensions must be installed:

- `php5-curl`
- `php5-gd`
- `php5-gmp`
- `php5-intl`
- `php5-json`
- `php5-mysqlnd`

Because not every Linux distribution makes all extensions available, you should check these dependencies before installing a local GNU social server.

For better server performance, the administrator can change some settings in the `php.ini` configuration file – for example, to enable cache memory for PHP. Before taking action, you will definitely want to consult the documentation.

Despite the already quite long development time of GNU social, it leaves something to be desired, especially if you plan to use a non-English version. If in doubt, I recommend using the instructions and wikis, along with the step-by-step instructions on how to set up your own GNU social server [18].

Mastodon

First released toward the end of 2016 under the AGPL, Mastodon [4] now has more than 1.5 million subscribers [19] and is expanding rapidly. The service is decentralized. Individual independent servers, or instances, each have their own user community. Managing the individual communities is the responsibility of the moderator of each instance. At the instance level, the admin also defines the access rights and content restrictions. The individual servers belong to the entire network of nodes.

The messages sent under Mastodon are called toots. They can be sent both

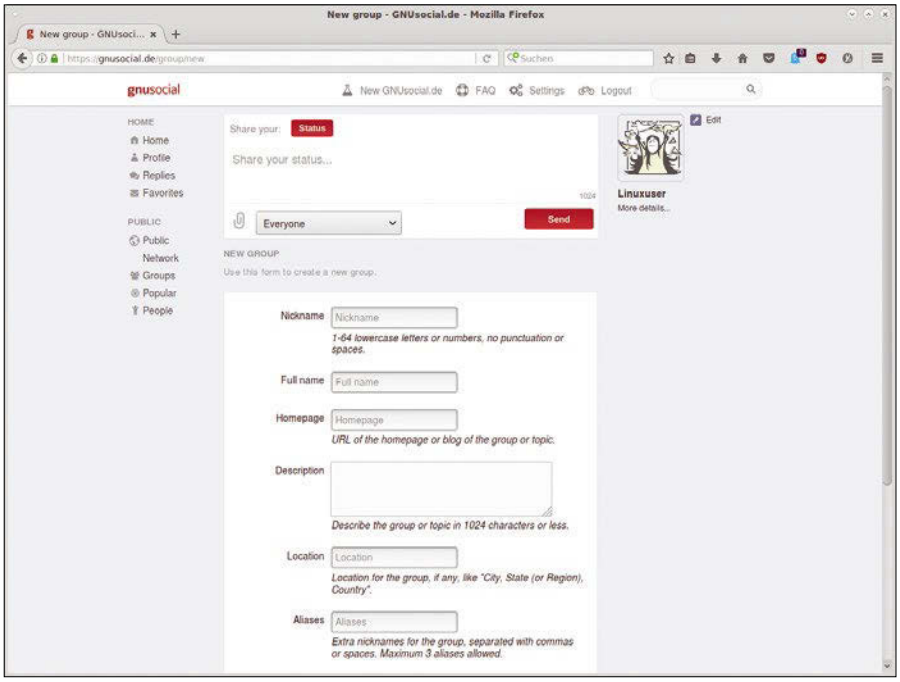


Figure 7: As the founder of a group, the user in GNU social is also its admin.

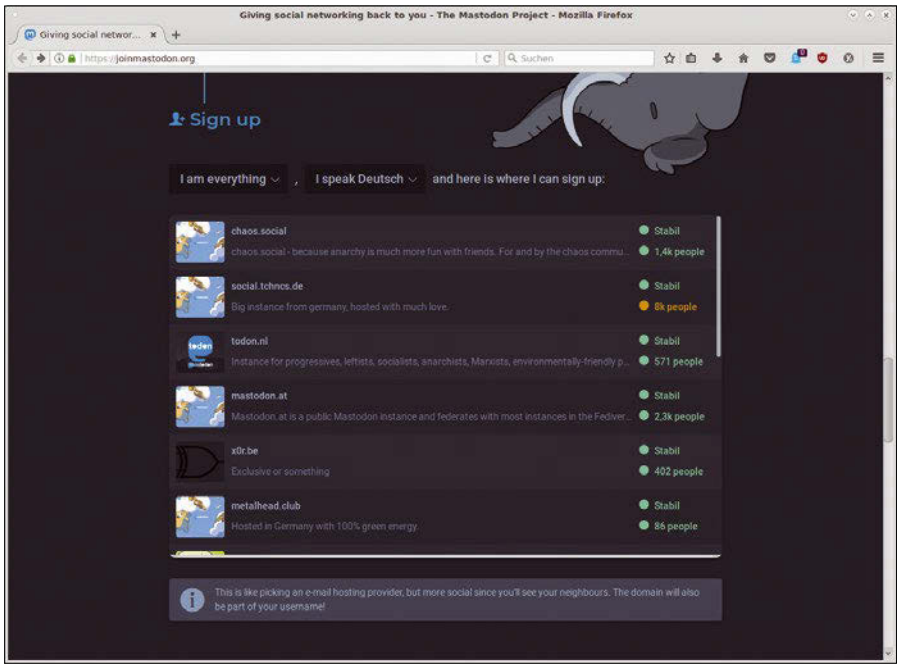


Figure 8: In Mastodon, a wizard first presents the individual available instances.

within and beyond the boundaries of a community (i.e., to users of other communities or to a network of communities). The same conditions apply to receiving toots.

Users can only make their messages available to individual users or groups. Messages can also be marked as private.

The free software servers are based on Ruby on Rails, and the front ends use JavaScript. Communication between the two is managed by the ActivityPub protocol [20].

Getting Started

To use Mastodon effectively, you need an overview of existing instances and their thematic orientation. For this, you should open the *https://joinmastodon.org* page in your web browser, which starts a wizard. The wizard first asks you to specify a preference and choose a language. It then displays a list of matching communities (Figure 8).

After clicking on an instance of interest, the wizard sends you to the main page of this instance, where you can gain an overview of the community and join.

When registering, Mastodon asks for a nickname, an email address, and a new password and sends you a confirmation email. The account is enabled by clicking on an embedded link; you can then log in with the specified access credentials.

The instance directs you to a clear-cut window that keeps track of all communications and local posts and where you can enter your own messages, with a maximum length of 500 characters. Of course, you can also forward and share posts under Mastodon.

In the right pane of the browser window, a dashboard lets you configure various settings (Figure 9), mainly with the *Settings* | *Preferences* item, where you can make important security settings. You can edit your own profile, enable two-way authentication (Figure 10), and import and export files with a smartphone. The files can be saved and loaded in different categories as CSV files, which you will need if you want to join other instances.

Because of the decentralized structure of the network, you can maintain a separate profile for each instance. Mastodon does not automatically transfer the contact lists and their rights settings from instance to instance. Once you have created a rights profile in a CSV file, you can import it into any new profile. In this way, you avoid having to specify blocked users or followers manually with the rights defined in each instance.

Own Instance

Companies with distributed locations might be interested in having their own Mastodon instance. The developers provide detailed documentation for this case [21]. Private users can also use this documentation to open and maintain a less restrictive instance. You don't even need your own dedicated server. Instance operators can offer the service through hosters, with Mastodon running on Ubuntu 16.04 LTS with Docker and a Nginx web server [22].

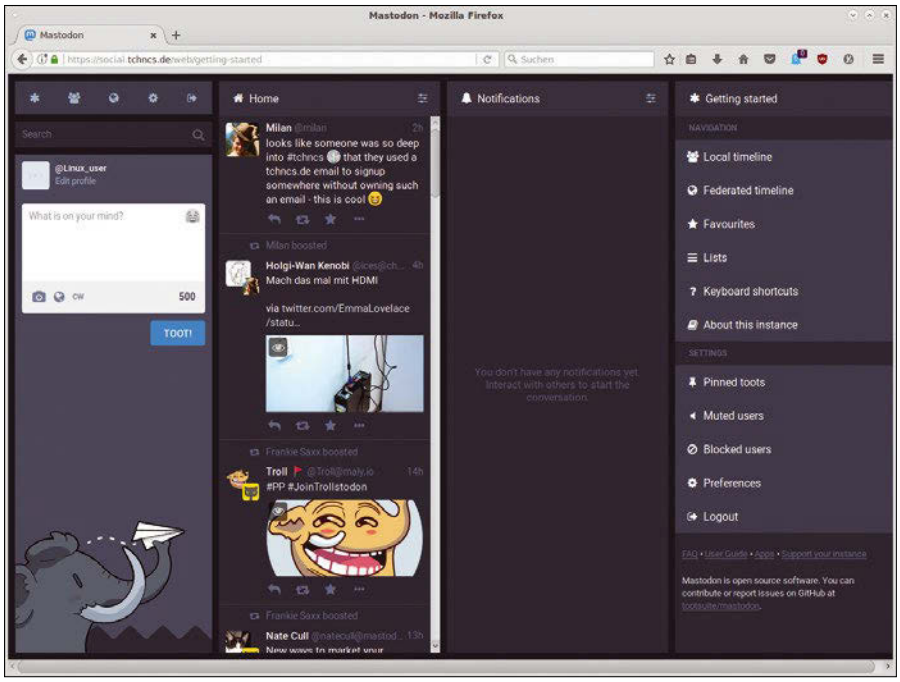


Figure 9: The Mastodon communication window is separated into different functional areas.

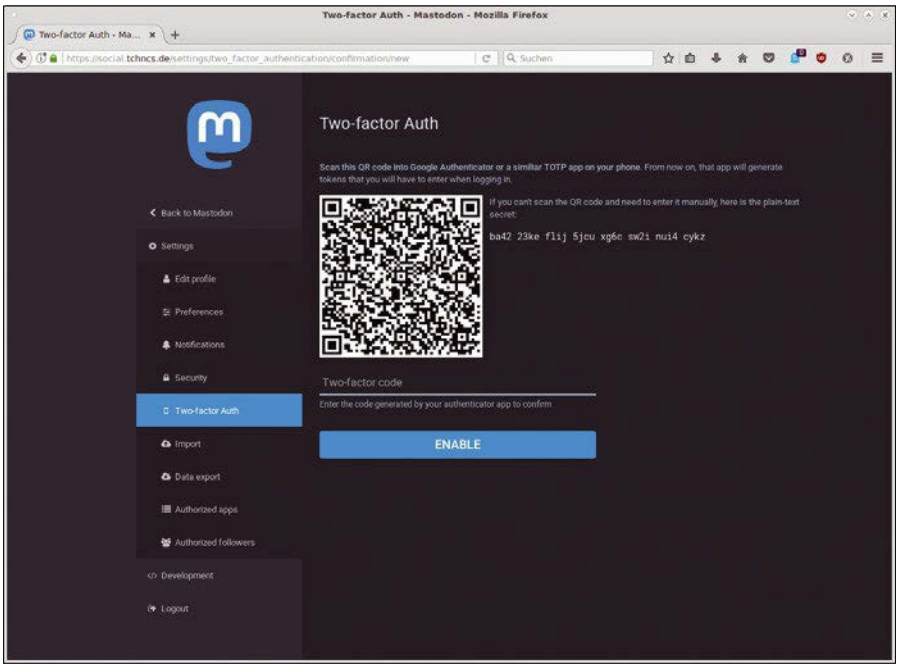


Figure 10: Mastodon users can use two-factor authentication.

The application server can also work separately from the content with different cloud services. All you need is your own domain, at least 2GB of RAM, and the corresponding free capacities on mass storage.

Companies can install the GitHub-hosted project on a dedicated server if they want to avoid cloud operation. Mastodon supports the SMTP protocol: The service sends email messages to administrators when certain events occur, so they always have an overview, even during intensive use of the platform.

Conclusions

None of the platforms tested need to hide their light under a bushel compared with the kings of the hill Facebook and Twitter (see also the "Other Social Networks" box). The services are stable and return control over their data to the operator, thanks to the pod concept and clear settings options. The user interfaces are also designed simply, without exception, and require practically no training.

The cumbersome and mostly very complex process for setting up the respective servers is a constant source of criticism. Occasionally, outdated documentation makes it even more difficult. Setting up your own server can therefore pose problems, even for experienced admins.

For companies, however, the small networks may be interesting because they enable closed groups in their own instances and guarantee the necessary security against unwanted eavesdroppers and co-readers. The only exception is Diaspora, whose developers do not want isolated "in-house installations" [5]. Nevertheless the pod admin still retains mastery over the data, with an option for ensuring privacy with rights assignments. ■■■

Other Social Networks

The spectrum of smaller social networks that exist independent of the large commercial applications, are decentralized, and are based on free software is much broader. However, the individual focus needs to be taken into account in each case. Discourse [23], for example, is already quite a widespread platform for interaction in forums. The program, distributed under the GNU GPLv2, is not designed as a short message service, but rather as a competitor to commercial applications like vBulletin.

Pump.io [7], which is free software for microblogging that emerged from the former Identi.ca, also relies on decentralized structures. Pump.io has numerous client applications for the desktop, but the software is still in an experimental stage, and features such as chat rooms are missing. WordPress, under development since 2003, is the market leader in the field of blogging software, and it can be transformed into microblogging software with the right theme [24].

Info

[1] Diaspora: *https://diasporafoundation.org*
[2] Friendica: *https://friendi.ca*
[3] GNU social: *https://gnu.io/social/*
[4] Mastodon: *https://joinmastodon.org*
[5] Diaspora and private pods: *https://wiki.diasporafoundation.org/FAQ_for_pod_maintainers#Can_I_make_my_pod_private.2Fisolated.2Fnot_communicate_with_other_pods.3F*
[6] Diaspora installation guides: *https://wiki.diasporafoundation.org/Installation*
[7] Pump.io: *http://pump.io*
[8] Choqok: *http://choqok.gnufolks.org*
[9] Friendica server list: *https://the-federation.info/friendica*
[10] Hotot: *https://github.com/lyricat/Hotot*
[11] Nginx: *https://nginx.org*
[12] Lighttpd: *https://www.lighttpd.net*
[13] StatusNet: *https://en.wikipedia.org/wiki/StatusNet*
[14] Group list for GNU Social: *https://gnu.io/social/try/*
[15] Heybuddy: *http://www.jezra.net/projects/heybuddy*
[16] Yaics: *https://stigatle.no/yaics/*
[17] Let's Encrypt: *https://letsencrypt.org*
[18] Setting up GNU social Server: *http://thomask.sdf.org/Social/en/admin/installing_gnu_Social.html*
[19] Details of user figures: *https://dashboards.mnm.Social/d/000000006/network-drilldown?refresh=30m&orgId=1&from=now-7d&to=now*
[20] ActivityPub protocol: *https://activitypub.rocks*
[21] Mastodon server installation: *https://github.com/tootsuite/documentation/blob/master/Running-Mastodon/Production-guide.md*
[22] Operating Mastodon with Docker: *https://github.com/ummjackson/mastodon-guide/blob/master/up-and-running.md*
[23] Discourse forum software: *https://www.discourse.org*
[24] WordPress theme P2: *https://wordpress.com/theme/p2*

Get the $200 of on registration by registering at www.snia.org/SDcode and using code SDC18LINUXPRO.

Don't Miss This!

Storage Developer CONFERENCE

Santa Clara, CA, September 24-27, 2018

SDC 18

Produced since 1998, SNIA's Storage Developer Conference (SDC) is the only event created **BY** storage developers **FOR** storage developers.

Register at www.snia.org/SDcode.

Register to attend SDC for technical discussions and education on the latest storage technologies and standards.

Register at www.snia.org/SDcode and use code SDC18LINUXPRO to recieve **$200 off** registration.

If you sell a product or service to storage developers, consider customizing a sponsorship package for SDC 2018.

Packages include speaking and exhibiting opportunities and group registrations.

www.storagedeveloper.org

SNIA®

Ogg Vorbis utilities

Supporting Role

With the addition of its supporting utilities, Ogg Vorbis competes with other free audio formats. *By Bruce Byfield*

Support for the free Ogg Vorbis audio format [1] is included by default in most Linux distributions. However, if you want to do more than play an Ogg file or rip a CD to the format using a desktop application, you need more than the Ogg codec. Although all-purpose tools are available for audio files, the most convenient tools for manipulating Ogg files are the format's supporting utilities, most of which are contained in the *vorbis-tools* package [2]. With these utilities, you can convert, play, and edit Ogg files and their metacomments from the command line.

Ogg Vorbis is an audio format that uses lossy compression, like MP3. Unlike FLAC, Xiph.Org's other free format that is popular with audiophiles for its high sound quality, Ogg has never been

```
ob@nanday:~/Downloads$ oggenc island-in-the-rain.wav
Opening with wav module: WAV file reader
Encoding "island-in-the-rain.wav" to
         "island-in-the-rain.ogg"
at quality 3.00
        [ 99.8%] [ 0m00s remaining] |

Done encoding file "island-in-the-rain.ogg"

        File length:  4m 28.0s
        Elapsed time: 0m 05.2s
        Rate:         51.1878
        Average bitrate: 108.2 kb/s
```

Figure 1: `oggenc` **converts several formats to** `.ogg` **and edits file comments.**

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. You can read more of his work at *http://brucebyfield.wordpress.com*

Photo by neil thomas on Unsplash

```
ob@nanday:~/Downloads$ ogg123 --device=pulse crest.ogg

Audio Device:   PulseAudio Output

Playing: crest.ogg
Ogg Vorbis stream: 2 channel, 44100 Hz
Album: Waiting for Bonaparte
Title: The Crest
Track number: 1
Compilation: false
Artist: Men They Couldn't Hang
Date: 1987
Genre: British Folk Rock
ReplayGain Peak (Track): 0.46352428
ReplayGain (Track): +2.32 dB
ReplayGain Peak (Album): 0.46352428
ReplayGain (Album): +2.50 dB
Time: 00:31.03 [03:15.66] of 03:46.69  (159.4 kbps)  Output Buffer  77.8%
```

Figure 2: `ogg123` **is a command-line music player.**

particularly popular. Many music players do not support it at all. Moreover, now that the MP3 format no longer has a restricted license, there is less reason for advocates of software freedom to use Ogg – although old habits die hard. However, there are still occasions where the smaller files created by lossy compression are useful, such as when using a player with limited memory, and at an equivalent sampling rate and file size, Ogg files offer higher sound quality than MP3s. Especially at their highest settings, Ogg files remain a reasonable compromise between file size and quality. By using the available utilities, you can get the most out of this often underrated audio format.

oggenc

The basic command-line tool for making that conversion from `.raw`, `.wav`, `.aiff`, or even `.flac` is `oggenc` (Figure 1). Ordinarily, `.flac` would be preferable to `.ogg` in terms of sound quality, but you might choose to create an `.ogg` file to produce a smaller file, especially one intended to be played where high sound quality would be wasted, such as on a bus or downtown.

Converting a file is as simple as typing the basic file followed by the file to convert, which will produce an `.ogg` file of medium quality with the same root name. Should you want a different name for the output file, add `--output=FILE`. You may also want more control over the quality of the output; you can achieve this by using the option `--quality=NUMBER`, where `1` is the lowest quality and `10` the highest. Similarly, if the output file is intended for playing on a mono system, you can use `--downmix` to reduce the quality of a stereo source file.

`oggenc` also offers considerable control over comments. You can eliminate comments altogether with `--discard-comments`. Or possibly you may prefer simply to edit comments with a series of options that take the structure `--comment=VALUE` and include such standard comments as artist, genre, date, track number, and title. You can also use `--lyrics=FILE`, which will display the words to a song in a music player like Amarok.

Still other options set the minimum and maximum bit rates, as well as other advanced options, about which average users are unlikely to care. See the `oggenc` man page for complete details.

ogg123

`ogg123` (Figure 2) is a playback command of considerable versatility. The simplest structure for the command is:

```
ogg123 --device=SOUNDSYSTEM INPUT-FILE
```

On most modern Linux systems, the sound system will be `pulse`, but if that doesn't work, then try `alsa` or `oss`. The

```
Processing file "crest.ogg"...

New logical stream (#1, serial: 74a6e980): type vorbis
Vorbis headers parsed for stream 1, information follows...
Version: 0
Vendor: Xiph.Org libVorbis I 20100325 (Everywhere)
Channels: 2
Rate: 44100

Nominal bitrate: 160.000000 kb/s
Upper bitrate not set
Lower bitrate not set
User comments section follows...
        ALBUM=Waiting for Bonaparte
        TITLE=The Crest
        TRACKNUMBER=1
        COMPILATION=false
        ARTIST=Men They Couldn't Hang
        DATE=1987
        GENRE=British Folk Rock
        REPLAYGAIN_TRACK_PEAK=0.46352428
        REPLAYGAIN_TRACK_GAIN=+2.32 dB
        REPLAYGAIN_ALBUM_PEAK=0.46352428
:
```

Figure 3: `ogginfo` **gives detailed information about the attributes of Ogg files, including all the metatags.**

```
bb@nanday:~/Downloads$ oggdec ./island-in-the-rain.ogg
oggdec from vorbis-tools 1.4.0
Decoding "./island-in-the-rain.ogg" to "./island-in-the-rain.wav"
        [100.0%]
```

Figure 4: `oggdec` **converts the formats of CDs and DVDs to** `.ogg`**.**

```
bb@nanday:~/Downloads$ vcut ./island-in-the-rain.ogg cut.ogg 30 +55
Processing: Cutting at 55.000000 seconds
```

Figure 5: `vcut` **produces truncated copies of** `.ogg` **files.**

input file is an `.ogg` or `.flac` file or a space-separated list of files.

However, that is only the beginning. You can create a playlist in which the paths to a file name are entered one per line and then use it with the `--playlist option`. You can use `--skipSECONDS` to set the start of playback and `--endSECONDS` to quit playback. With `--repeat`, the files specified will play again from the first one. Alternatively, `--shuffle` will play a list of files once in random order, while `--random` will play the list of files randomly over and over. If you are having trouble with playback, the option `--buffer=KILOBYTES` may improve performance.

ogginfo

Vorbis files contain a wealth of information about themselves, and `ogginfo` (Figure 3) displays it all, starting with bit rates, moving through comments, going on to the data length or file time and playback time in minutes and seconds, and ending with the average bit rates.

For convenience, you can query multiple files at one time by specifying files in a space-separated list after the basic command. However, even a single file returns more than a screenful of information on anything except the largest screens, so you should probably pipe `ogginfo` through the `less` command so that the output is more manageable.

oggdec

`oggdec` (Figure 4) is a simple converter between file formats. The command structure is:

```
oggdec FILE.ogg
```

Multiple output files can be specified with a space-separated list, and output files are created using original file names for the name and adding a different extension. If you are importing a single file, you can use `--outputfile=File` to change the file name. The default output is a `.wav` file, the format usually used on a CD.

The option `-bits=NUMBER` can be used to specify an 8- or 16-bit file. You can also create an output in RAW format by specifying `--raw` as an option. For 16-bit files, you can also specify endianness with the option `--endian=`, using `1` for small-endian and `0` for big-endian, or signedness with `sign=`, using `0` for unsigned and `1` for signed. However, today usually you should not need to consider either of these optons.

vcut

`vcut` (Figure 5) creates copies of an existing `.ogg` file, beginning and ending at the specified locations, which are designated in seconds. The format is:

```
vcut SOURCE OUTPUT START-POINT +ENDPOINT
```

The original file is not affected.

```
bb@nanday:~/Downloads$ vorbiscomment -l ./island-in-the-rain.ogg
ALBUM=Waiting for Bonaparte
ARTIST=Men They Couldn't Hang
COMPILATION=false
DATE=1987
GENRE=British Folk Rock
TITLE=Island in the Rain
TRACKNUMBER=5
REPLAYGAIN_TRACK_PEAK=0.44521883
REPLAYGAIN_TRACK_GAIN=+2.72 dB
```

Figure 6: `vorbiscomment` **is an editor for the metatags of Ogg files.**

```
bb@nanday:~/Downloads$ vorbisgain *.ogg
Analyzing files...

    Gain    |   Peak   | Scale | New Peak | Track
------------+----------+-------+----------+------
 +2.32 dB   |   15188  |  1.31 |    19839 | crest.ogg
 +8.17 dB   |   10523  |  2.56 |    26955 | cut.ogg
 +2.82 dB   |   15413  |  1.38 |    21324 | island-in-the-rain.ogg
```

Figure 7: `vorbisgain` **smooths out the volume levels of both albums and individual files.**

vorbiscomment

Music players generally include options for displaying files by categories such as artist, title, track number, and genre. `vorbiscomment` (Figure 6) is a utility for editing those comments and adding new categories. These comments are usually ripped along with the file and can be edited or expanded as you choose.

To see the existing comments, enter:

```
vorbiscomment --list FILE.ogg
```

A number of comments are standard to all files, but you can add your own with the options:

```
--append --comment 'NAME=VALUE'
```

Alternatively, you can place all the comments and modifications you want to make in a file, one per line, and add them to the file using the options `--append --comment`. Use the option `--append FILE.ogg`, press Return, add a comment in the format `NAME=VALUE` (one per line), and press Ctrl + D when finished. Existing comments can be replaced by using the `--write` option alongside the `--append` and `--comment` options. Comments can be written in UTF-8 format by the addition of `--write`.

vorbisgain

`vorbisgain` (Figure 7) is not packaged with *vorbis-tools*, but it is a related tool that you might want to download at the same time [3]. The utility is intended to maintain a uniform sound or replay gain across files on the same album.

To begin using `vorbisgain`, enter the command

```
vorbisgain --album ./*.ogg
```

adding the `--recursive` option if some of the files to modify are in subdirectories. The command returns the settings for each file with the same directory comment and a recommended setting that is usually the average of all the files. The gain on each file can then be set with the option `--album-gain=SETTING`. You may want to preserve file attributes with the `--preserve-m-time` option. Should the results not to be to your liking, you can use `--clean` to remove the gain comments created by `vorbisgain` before making adjustments.

Adding Control

When these utilities are added to a system, the Ogg Vorbis format becomes a much more attractive alternative. In particular, `ogg123` is a compact music player, while, aside from its specialization, `vorbiscomments` is one of the most versatile editors for meta-comments that I have found. As for `vorbisgain`, I can think of no other audio format for which a similar command-line tool is available. You may not always need all the available Ogg utilities, but they make the format more of an option than it is by itself. ■■■

Info

[1] Ogg Vorbis format: *https://en.wikipedia.org/wiki/Vorbis*

[2] vorbis-tools: *https://github.com/xiph/vorbis-tools*

[3] vorbisgain: *https://sjeng.org/vorbisgain.html*

A Python script calculates the solution to the Chinese Rings puzzle

Time Waster

Mike Schilli takes on the almost 2,000-year-old Chinese Rings puzzle. Instead of just jingling rings, he tries to find a solution with logical operators. *By Mike Schilli*

I am not a fan of puzzles due to my lack of patience. However, when I recently read in an excerpt from Martin Gardner's venerable 1972 *Scientific American* column [1] in an online antiquarian bookstore that the Chinese Rings puzzle could be solved with Gray codes [2] from the field of information theory, I was gripped by game fever. I ordered the ring set online for little money.

A second-century Chinese general named Zhuge Liang is said to have invented the game, which was nicknamed "Baguenaudier" [3] (time waster) many centuries later. Allegedly, his intention was to keep his wife busy during his absence. The metal contraption arrived in a cardboard box with printed Chinese characters. Exhibiting great foresight, I immediately clamped the rail with the silver rings in my vise for electronic crafts to prepare for some time-consuming tinkering.

The nine inconspicuous rings initially all sit on two metal rails connected in the front, and they are also tied to one another through small metal rods (Figure 1). This restrictive suspension initially gives the impression that nothing can be changed at all in the entire construction, but the enclosed operating instructions indicate that there are indeed a limited number of possible moves.

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.

Ticket to Ride

The player moves one ring per turn by guiding it through the middle gap of the rail, either to lift it onto the rail or to remove it from there and move it down through the rail opening. The game is subject to precisely two restrictions: The first (outer right) ring can be moved freely at any time. Any other ring can only be moved if (a) its direct right-hand neighbor is up on the rail and (b) all other rings to its right are down.

In the initial constellation in Figure 1, two moves are possible: The player can pull off the first ring to the right, lift it up, skew it, and then let it drop down through the central opening in the rail. If the player leaves the first ring alone, the second ring on the rail can be considered as an alternative. Since the second ring from the right has only one ring to the right (ring number 1), which is also at the top of the rail, the former can be moved downward.

In the second case, the player pushes the right ring a little further to the right to the end of the rail (without dropping it) and at the same time pulls the second ring to the right, guiding it to the right out of the rail, then skewing it, and pushing it down through the rail gap.

The same is true for moving a ring from bottom to top; in Figure 2 ring 4 is down, while ring 3 is up, and rings 2 and 1 are down. According to the rules, the player can lift ring 4 up through the gap in the rail, guide it to the right past ring 3 at the edge of the rail, thread it from the right onto the rail, and then deposit it (Figure 3).

As you can read on Wikipedia [3], all nine rings of the puzzle can be removed in a total of 341 moves, so that in the end – surprisingly – only the empty rail remains. The absolutely annoying thing about the procedure is that the player has to backtrack dozens of times, because to remove ring 9, for example, ring 8 must be at the top, but rings 1 to 7 must be at the bottom.

How does ring 7, which is initially on top like all other rings, reach the bottom? With ring 6 at the top, while rings 1 to 5 are at the bottom. How does ring 5 reach

Figure 1: At the start, all nine rings sit on the metal frame.

Figure 2: Ring 3 is at the top; rings 1 and 2 are at the bottom, so ring 4 …

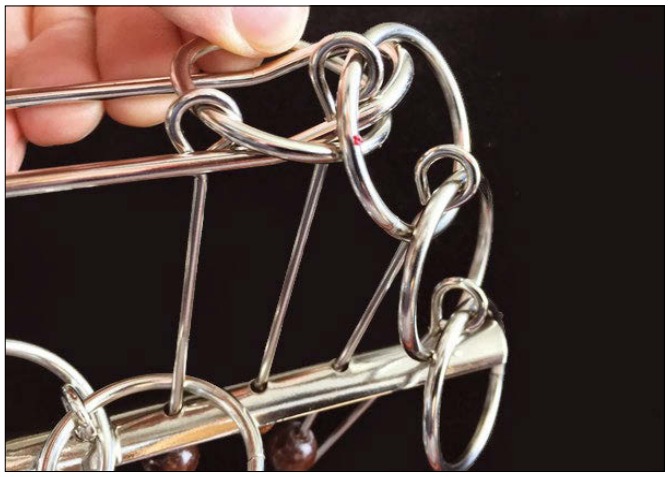


Figure 3: … can be pulled up and placed at the top through ring 3.

Listing 1: graycode.py

```
01 #!/usr/bin/env python3
02 import operator
03
04 def grayme(num):
05     shifted=num>>1
06     return(operator.xor(num,shifted))
07
08 def main():
09     for i in range(15):
10         print(i, format(i,"08b"),
11               format(grayme(i),"08b"))
12
13 if __name__ == "__main__":
14     main()
```

the bottom? With ring 4 at the top, while rings 1 to 3 are at the bottom.

And so it goes, back and forth, until finally ring 9 is at the bottom and then ring 8 – until finally ring 1 drops off after the 341st move. In general, the formula for the minimum number of moves for odd numbers is

$$\left(2^{n+1}-1\right)/3$$

increasing exponentially with the number of rings.

The player has to think carefully about which ring to turn next; a move in the wrong direction means you have to turn around later on and retrace your steps, because otherwise you will end up going round in circles instead of making progress.

Let's Automate!

At first glance, the repetitive moves with the rings might remind a mathematically inclined person of binary numbers, but those tend to change by more than one bit at a time. Just think of the sequence going from `0111` (5) to `1000` (6), where four bits (or rings) change at the same time. Gray codes [2] behave differently and change only by one bit at each step. Instead of `00`, `01`, `10`, `11`, Gray code, which is named after the physicist Frank Gray, counts `00`, `01`, `11`, `10`, and luckily, there's a simple formula to convert binary numbers to Gray code [3]:

```
num XOR (num >> 1)
```

According to the formula, the number 2 (binary `10`), for example, becomes `11` thanks to the one-bit shift to the right, and the XOR operator (^) connects `10` and `01` to create `11`, as it always returns a true bit if the bits of both operators differ at one point. Listing 1 [4] implements it in a simple Python script with the `grayme()` function. It gets the XOR operator from the `operator` package as the `xor()` function.

As a practical test, the `for` loop in the `main()` function iterates from line 8 on, over the numbers from 1 to 14, and outputs the number in each round in decimal, binary, and Gray code (Figure 4).

Contrary to its philosophy ("There's one way to do it"), Python offers three different methods for string formatting à la `printf()`. Listing 1 uses the core function `format()`, which outputs integers with the format string `08b` in 8-bit width as binary numbers with leading zeros. The `print` statement also outputs the decimal number `i` itself, as well as the Gray code generated with `grayme()` as binary bits.

Command or Library

The typical Python code snippet

```
if __name__ == "__main__"
```

```
$ ./graycode.py
0 00000000 00000000
1 00000001 00000001
2 00000010 00000011
3 00000011 00000010
4 00000100 00000110
5 00000101 00000111
6 00000110 00000101
7 00000111 00000100
8 00001000 00001100
9 00001001 00001101
10 00001010 00001111
11 00001011 00001110
12 00001100 00001010
13 00001101 00001011
14 00001110 00001001
$
```

Figure 4: Gray codes of the numbers 1 to 14 only differ by one bit from their predecessors.

checks if the script was called at the command line and, in this case, jumps to the `main()` function as of line 8. However, if the script was pulled into another script as a package using `import graycode`, it does not execute the `main()` code, but integrates the `grayme()` function into the script, which can then invoke it by calling `graycode.grayme()`.

Alternatively, a Python script can import the function directly into its namespace using

```
from graycode import grayme
```

so that `grayme()` simply works there.

Trial Run

At first glance, the Gray code in Figure 4 seems to be a workable solution to the ring problem. When the bits in a particular state are all `0`, all rings are off the rail; if they are all 1, they're all on top, so the generated sequence shows a solution to get from the "all off" state back to the puzzle's initial state when it arrived in the mail. To go from the initial "all on top" state to "all off," the puzzle master has proceed backward in the generated sequence.

Is the procedure correct in detail, though? I'll whip up a test script to run through all the codes and check the two conditions of the game for each one: Is only one ring really moved at a time, either from top to bottom (`1=>0`) or from bottom to top (`0=>1`)? And does the autoplayer only move the first ring, or, if it is another ring's turn, is its right neighbor on top as required, and are all the other rings down? Time to dissect the Gray code with Python's bit operations.

The utility function `bits_set()` starting in line 10 of Listing 2 searches through the bits of a number and returns an array with the index numbers of the bits set to 1. Among other things, it serves as a controller to verify that I've arrived at the end of the sequence when all nine rings are on top, that is, when the first nine-element array with index numbers comes back from `bits_set()`.

The test script also checks how many and which bits have changed from one Gray code to the next and ensures that only one bit (i.e., one ring) has moved each time. To do this, the `rings_changed()` function processes two Gray codes from line 6 with the XOR operator, which sets bits to 1 at the positions that differ.

Listing 2: test.py

```
01 #!/usr/bin/env python3
02 import operator
03 import math
04 from graycode import grayme
05
06 def rings_changed(pos1,pos2):
07     xor = operator.xor(pos1,pos2)
08     return bits_set(xor)
09
10 def bits_set(num):
11     bits = []
12     while True:
13         if num == 0:
14             break
15         bit = int(math.log(num,2))
16         mask = 1 << bit
17         num  &= ~mask
18         bits.append(bit)
19     return(bits)
20
21 def move_valid(pos1,pos2):
22     changed=rings_changed(pos1, pos2)
23
24     if len(changed) != 1:
25         # only one ring can change per move
26         raise Exception(
27           "More than one change: " +
28           str(changed))
29
30     if changed[0] != 0:
31         # next ring up?
32         mask = 1 << changed[0]-1
33         if not (pos2 & mask):
34             raise Exception(
35               "Next ring not up: " +
36               str(changed))
37
38         if changed[0] > 1:
39             # right-most rings down?
40             mask = ~(~0 << changed[0]-1)
41             rest = pos2 & mask
42             if len(bits_set(rest)) != 0:
43                 raise Exception(
44                   "Rings not down: " +
45                   str(changed))
46 def main():
47     i=0
48     rings=9
49     last_gray=None
50     while True:
51         gray=grayme(i)
52         print(format(gray,"08b"))
53
54         if last_gray is not None:
55             move_valid(last_gray, gray)
56
57         last_gray=gray
58
59         if len(bits_set(gray)) == rings:
60             break
61
62         i += 1
63
64 if __name__ == "__main__":
65     main()
```

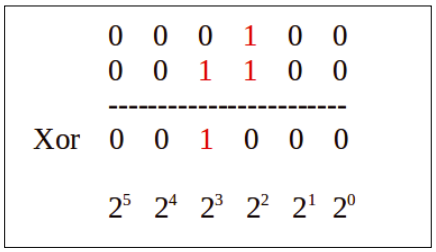


Figure 5: Using XOR, the test script checks which rings the user moves according to the Gray code.

Then the loop counts the 1 bits in the endless `while` loop in line 12 and appends their indices to the array returned by the function, using `append()` (Figure 5). For this to happen, the loop determines the index of the highest set bit in the current state with the formula

```
int(math.log(num,2))
```

which calculates the logarithm of the number to be examined to base 2 and rounds it down to the next integer value. To clear the highest bit found this way, it then generates a mask with a bit set at the critical position and combines the number and the mask with a bit-wise AND operator:

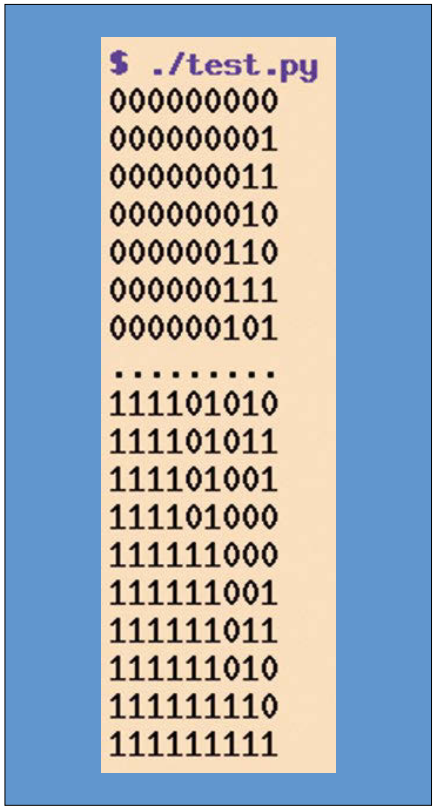


Figure 6: The test script checks 341 Gray code transitions for their validity in the puzzle.

```
mask = 1 << bit
num  &= ~mask
```

The next round of the loop may find more bits in the lower ranks, register them, and clear them, until the value finally turns to `0` and the loop is aborted.

Whether a move proposed by the Gray code is valid is checked by the `move_valid()` function starting in line 21. First, it throws an exception if more than one bit (ring) has changed (line 26), so the main program does not have to check its return value at all and simply relies on the fact that it will blow up if something is off. For rings 2 and above, the `if` construct as of line 30 uses one mask to check whether all right-hand side rings are on top and otherwise terminates with an error in line 34.

For rings 3 and above, Lines 38 and following use a mask of 1 bits to check whether all rings further on the right are at the bottom.

Right-Aligned 1s

A mask with *n* right-aligned 1s is generated by the following somewhat strange construct

```
~(~0 << n)
```

in line 40. The `~0` first creates the bit complement of the number `0` (i.e., an integer in which all bits are set to 1). It then shifts the bits by `n` digits to the left, which fills the first *n* bits on the right with `0`s, while the rest of the number keeps 1 bit values.

Another complement `~` outside the parenthesis flips all bits again, so that the mask has set the first *n* bits on the right to 1 as desired. Figure 6 finally shows the output of the test script and verifies that the Gray code complies with all rules of the ring game.

The bit-wise method is not suitable for a larger numbers of rings, but works up to the integer limit of 32 or 64 bits, depending on the platform. Ring games with more than 10 rings are not practical anyway, because the number of combinations would skyrocket. The game is a nice way to pass the time (Figure 7) but can result in a nervous breakdown. It is important to plan the moves to the next ring to be removed. It is incredibly frustrating if you remove a ring that needs to stay on top in order to remove another ring in the heat of the moment. This means retracing all your steps and wasting even more time. ■■■

Info

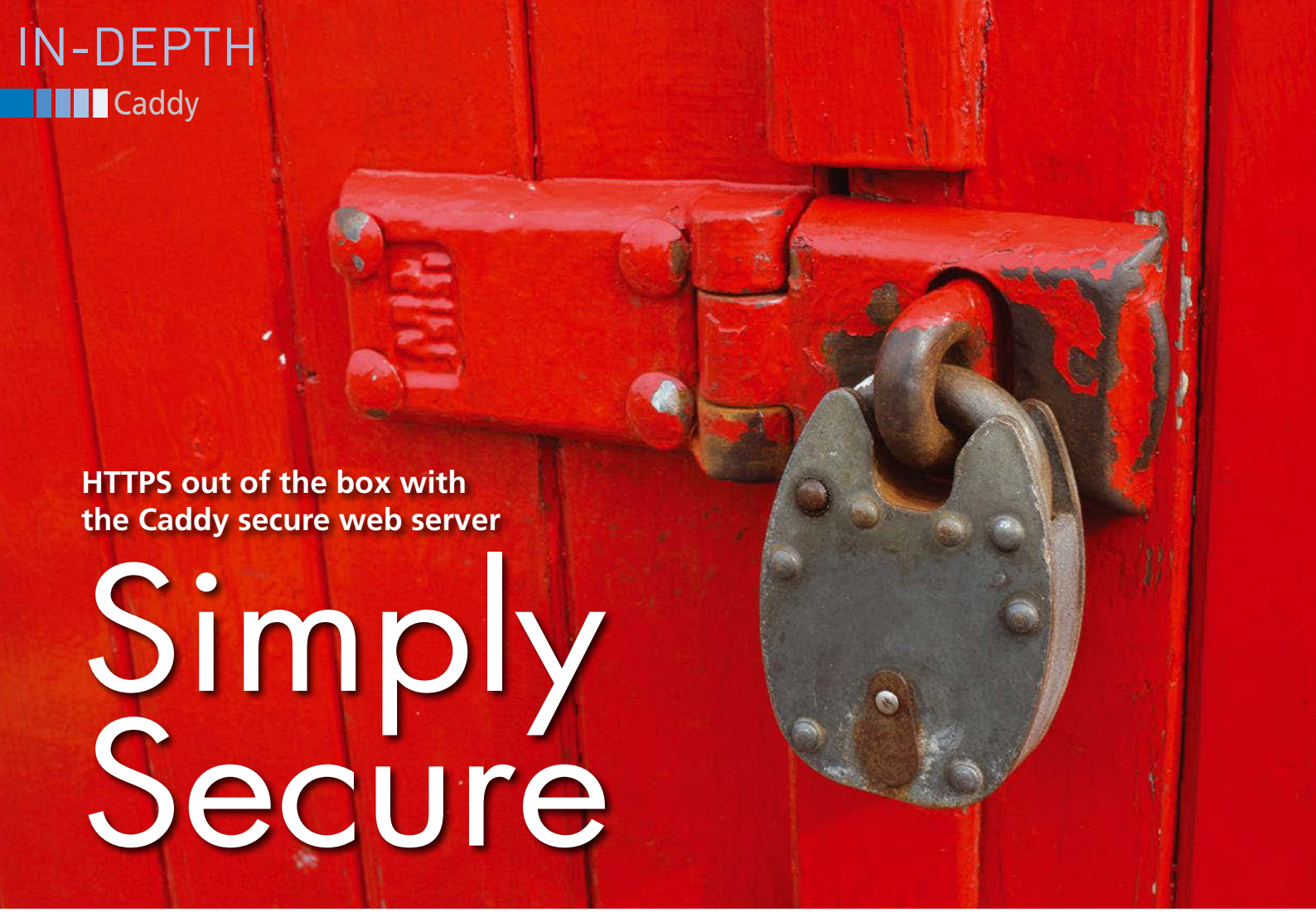
[1] *Martin Gardner: The Magic and Mystery of Numbers*. Scientific American, 2014: *https://www.scientificamerican.com/store/books/martin-gardner-the-magic-and-mystery-of-numbers/*

[2] Gray code: *https://en.wikipedia.org/wiki/Gray_code*

[3] Baguenaudier: *https://en.wikipedia.org/wiki/Baguenaudier*

[4] Listings for this article: *ftp://ftp.linux-magazine.com/pub/listings/linux-magazine.com/213/*

Figure 7: After about 250 moves, just before I solved the puzzle.



HTTPS out of the box with the Caddy secure web server

Simply Secure

Caddy lets even the most inexperienced user set up a secure web server.

By Frank Hofmann and Mandy Neumeyer

After hundreds of high-profile attacks on large and small websites, the web world is gradually giving up on old-fashioned, unsafe, unencrypted HTTP and is moving to the safer, encrypted variant known as HTTPS. HTTPS is based on the SSL/TSL protocols, which means you need to install, maintain, and regularly renew digital certificates for the website. The certificates are only valid for a limited period, which can vary between three months to 10 years, depending on the certificate authority.

All the major web servers, such as the Apache HTTP server, Microsoft's Internet Information Services (IIS), and Nginx, provide some means for HTTPS support, but the configuration steps are often complicated and more trouble than you need, especially if you just operate a small website.

Caddy [1] is a simple and easy web server that delivers simple and convenient HTTPS support. Caddy uses the Let's Encrypt [2] project for easy access to free digital certificates. (See the box titled "Simply Encrypted.") Caddy also includes support for HTTP/2 (see "From HTTP 1.1 to HTTP/2"), and it comes with other useful features, such as support for the Markdown and AsciiDoc formatting languages.

Simply Encrypted

The Let's Encrypt project is the first major initiative of the non-profit organization Internet Security Research Group (ISRG). The goal of the ISRG is to make encrypted WWW connections the standard; in other words, HTTPS for everyone. Let's Encrypt uses domain validation certificates for encryption. To check whether the person submitting a certification request actually owns the domain, Let's Encrypt creates and sends a token to the requesting domain. This step is followed by a web or DNS query, in which the domain is verified using a key derived from this token. This ensures that the content actually originates from the operator of the website. Let's Encrypt also offers the greatest possible transparency for its own trustworthiness. The project uses free software and open standards, in combination with the regular publication of transparency reports. To date, the project has already issued certificates for 63 million domains.

From HTTP 1.1 to HTTP/2

Version 2.0 of HTTP [3] was released in February 2015. HTTP/2, which is described in RFC 7540 [4], has been gradually replacing HTTP 1.1 since 1997. Evaluations of WWW data traffic revealed a 20 percent share of HTTP/2 traffic at the end of 2017. HTTP/2 comes with a number of innovations, including data compression of the HTTP headers and parallel execution of requests with a single TCP connection ("pipelining"). In general, the protocol aims to make better use of the available computing power and to minimize the time required to transmit a request.

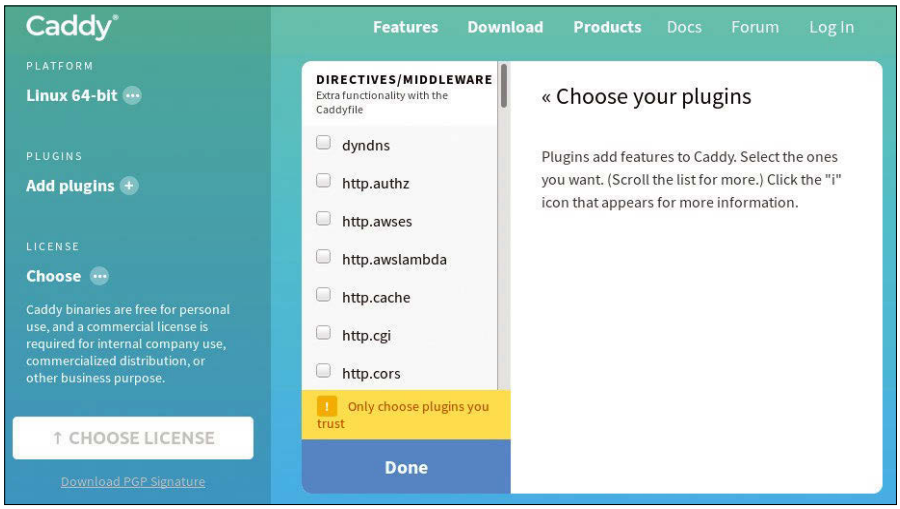


Figure 1: On the project page, you define which extensions you want to include with the software.

Caddy is written in the Go programming language, and it is available as a free community version for private use. If you want to use the web server commercially, the developers would be happy to receive a one-off payment of $25. The project is licensed under Apache 2.0.

This article is adapted to Debian 9 and is partly based on a blog post by Supriyo Biswas [5].

Installation

Caddy is available for Linux, Windows, many BSD variants, and Mac OS. The Caddy developers do not provide separate versions for different Linux distributions. In the download area of the website, a configurator helps you select the appropriate binary for your system, as well as plugins you might want to integrate (Figure 1). The software then generates a package to suit your needs.

If you prefer to set up Caddy at the command line, say, for use on a headless system, first pick up a shell script from the Caddy project website (Listing 1, line 1), make it executable (line 2), and download the desired software image (line 3) by calling the script. You can specify additional plugins in the call as comma-separated parameters (line 4). Table 1 shows a small selection of available extensions.

Basic Setup

To test the functionality, first create an `index.html` file in the current directory; this file consists only of the HTML header and the body "Hello World." Then, launch Caddy by typing `./caddy` in the same directory. To access the page in the web browser, use either the IP address of the host or *localhost* or *127.0.0.1* and add port number *2015*, on which Caddy listens for HTTP requests without any further settings (Figure 2).

To secure Caddy and set it up as a system service, create a separate account for Caddy, then assign it a home directory of `/opt/caddy/` with the `-rmd` option (Listing 2, first line). This directory also contains the configuration files and the actual content of the website.

You'll need to create the three subdirectories `store`, `logs`, and `web` (Listing 2, second line); `store` contains – among other things – the SSL certificates, `log` contains the logfiles for the web server, and `web` contains the actual website. To make sure that Caddy can access it, assign the three directories to the previously created *caddy* account (Listing 2, last line).

Next, copy the previously created `index.html` file to `/opt/caddy/web/`. Caddy now needs some help in the form of a configuration file to find the contents. You can choose the name of the configuration file – in this article, the file is called `/opt/caddy/caddy.config`. Enter the contents from Listing 3, which is the

Listing 1: Get Caddy

```
# wget https://getcaddy.com -O getcaddy
# chmod +x getcaddy
# ./getcaddy personal
# ./getcaddy personal http.ipfilter,http.cache
```

Table 1: Caddy Plugins

`dyndns`	Dynamic DNS, for example, via Cloudflare
`http.cache`	Caching for HTTP
`http.cgi`	Extension with CGI scripts
`http.expires`	Setting an expiration date
`http.git`	Publication/updates via `git` commands
`http.ipfilter`	Block access by selected IP addresses
`http.webdav`	Extension for WebDAV

Listing 2: Securing Caddy

```
# useradd -rmd /opt/caddy caddy
# mkdir /opt/caddy/{store,logs,web}
# chown -R caddy: /opt/caddy
```

Listing 3: Configuring a Website

```
01 http:// {
02   root /opt/caddy/web/default
03   log /opt/caddy/logs/default.log
04   gzip
05 }
```

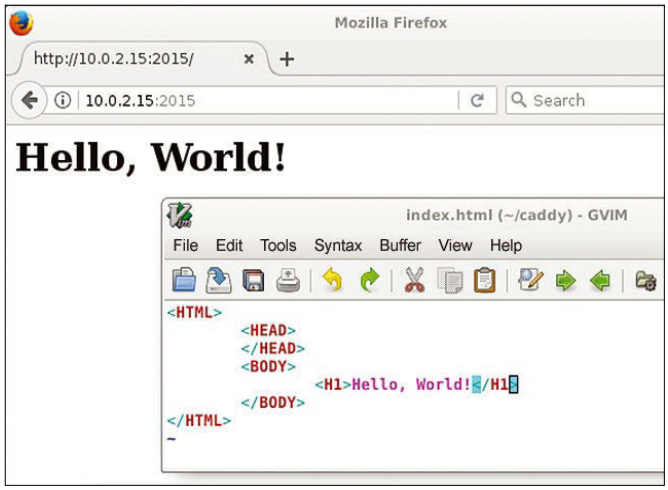


Figure 2: The website for the functionality test and the call in the web browser.

configuration for a single website on the local computer.

Line 1 is for serving up HTTP content on port 80; line 2 defines the root directory of the website, and line 3 defines the corresponding logfiles. Line 4 contains the Gzip module, which ensures that Caddy delivers compressed HTML and text files. With the performance of today's systems, compression causes only a minimal delay, which more than compensates for the transmission time saved.

If you want to provide multiple web services on the system, you need to create a separate block for each service. In Listing 4, the third code block describes a web server that delivers its data via port 8080. All you need to do is enter `:8080` after the domain name.

Now you need to set up Caddy as a regular service. For systems with a systemd, create a separate entry in the `/etc/systemd/system/caddy.service` file (Listing 5). In the `Unit` section, name the service and enter a description. The `Service` section defines the user and group, the environment variables, the start parameters, what happens when the process is restarted, and the limits for the number of open files and parallel processes. The `Install` section determines the target or runlevel for which the service is available.

Use the first two commands from Listing 6 to start the previously configured service and check whether it is running correctly (Figure 3). Then, visit the website and check its accessibility. This time Caddy is no longer listening on port 2015, but on the usual port 80 for HTTP.

If you change the configuration at a later time, you need to restart the service, as shown in the last line of Listing 6.

Listing 4: Multiple Sites

```
http:// {
  root /opt/caddy/web/default
  log /opt/caddy/logs/default.log
  gzip
}

http://out-of-space.example.net {
  root /opt/caddy/web/out-of-space
  log /opt/caddy/logs/out-of-space.log
  gzip
}

http://on-the-road.example.net:8080 {
  root /opt/caddy/web/on-the-road
  log /opt/caddy/logs/on-the-road.log
  gzip
}
```

Listing 5: caddy.service

```
[Unit]
Description=Caddy HTTP/2 web server

[Service]
06 User=caddy
Group=caddy
Environment=CADDYPATH=/opt/caddy/store
ExecStart=/usr/local/bin/caddy
   -agree=true
   -log=/opt/caddy/logs/caddy.log
   -conf=/opt/caddy/caddy.config
   -root=/dev/null
ExecReload=/bin/kill -USR1 $MAINPID
LimitNOFILE=1048576
LimitNPROC=64

[Install]
WantedBy=multi-user.target
```

Switching to HTTPS

Caddy impresses with its integration of Let's Encrypt for secure HTTPS connections. To add Let's Encrypt, you must have a domain. In the second block of Listing 4, change `http` to `https` in the first line. Then add line 5 to the block (Listing 7). Let's Encrypt informs you about changes in the certificate for HTTPS, especially before it expires and needs to be renewed.

You do not need an explicit statement for HTTP. All requests for this protocol will automatically switch to HTTPS on port 443. Now restart Caddy, and it will automatically connect to Let's Encrypt and set up a certificate.

Content as Markdown

The Markdown [6] or AsciiDoc description languages provide additional options for content creation in Caddy. Caddy interprets Markdown natively.

You'll need to modify the configuration file to use Markdown with Caddy. Listing 8 contains the `markdown` entry in lines 5 to 7, followed by a / and a

Listing 6: Start the Service

```
# systemctl enable caddy.service
# systemctl status caddy.service
# systemctl restart caddy.service
```

Listing 7: Adding Let's Encrypt

```
https://out-of-space.example.net {
  root /opt/caddy/web/out-of-space
  log /opt/caddy/logs/out-of-space.log
  gzip
  tls admin@out-of-space.example.net
}
```

```
● caddy.service - Caddy HTTP/2 web server
   Loaded: loaded (/etc/systemd/system/caddy.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2018-01-07 15:38:41 CET; 1s ago
 Main PID: 1164 (caddy)
    Tasks: 6 (limit: 4915)
   CGroup: /system.slice/caddy.service
           └─1164 /usr/local/bin/caddy -agree=true -log=/opt/caddy/logs/caddy.log -conf

Jan 07 15:38:41 debian-capetown systemd[1]: Started Caddy HTTP/2 web server.
Jan 07 15:38:41 debian-capetown caddy[1164]: Activating privacy features... done.
Jan 07 15:38:41 debian-capetown caddy[1164]: http://
```

Figure 3: The systemd status display tells you that Caddy is doing its job.

Listing 8: Markdown and Plain Text

```
01 http:// {
02   root /opt/caddy/web/default
03   log /opt/caddy/logs/default.log
04   gzip
05   markdown / {
06     ext .md .txt
07   }
08 }
```

TIP

If you are interested in a secure connection for other sites that you visit in everyday life, HTTPS Everywhere is worth a look: This free software, which was developed by the Electronic Frontier Foundation (EFF) and the Tor project, automatically switches from unsecure HTTP transmission to encrypted HTTPS transmission – even if you have accessed HTTP. A green padlock in the address line indicates the change to secure transmission. However, many websites also implement content from third parties that might not use HTTPS. HTTPS Everywhere is available for Mozilla Firefox [8] and Google Chrome [9].

block in curly brackets. Caddy interprets the Markdown permission for the entire website. In line 6, you define which name extensions Caddy accepts, in this example, Markdown (`.md`) and plain text (`.txt`). Additionally, Caddy processes information about CSS and JavaScript, as well as individual templates [7].

Conclusions

In our tests, Caddy was very fast and reliable, and its modular design makes it quite flexible and easy to configure. Setting up a secure web server has never been easier, and a massive benefit is the ability to serve up encrypted websites. We give Caddy five out of five stars. ■■■

Authors

Frank Hofmann works on the road – preferably from Berlin, Geneva, and Cape Town – as a developer, trainer, and author. He is also the co-author of the Debian package management book (*http://www.dpmb.org*).

Mandy Neumeyer has lived in Cape Town for nine years and likes to travel around the world. She works in tourism and is currently building up an extra source of income as a digital nomad.

Info

[1] Caddy: *https://caddyserver.com*

[2] Let's Encrypt: *https://letsencrypt.org*

[3] HTTP/2: *https://en.wikipedia.org/wiki/HTTP/2*

[4] RFC 7540: *https://tools.ietf.org/html/rfc7540*

[5] "How to Host a Website with the Caddy Web Server on Linux": *https://www.booleanworld.com/host-website-caddy-web-server-linux/*

[6] Markdown: *https://linuxhint.com/introduction-to-markdown/*

[7] Markdown in Caddy: *https://caddyserver.com/docs/markdown*

[8] HTTPS Everywhere (Firefox): *https://addons.mozilla.org/de/android/addon/https-everywhere/*

[9] HTTPS Everywhere (Google Chrome): *https://chrome.google.com/webstore/detail/https-everywhere/gcbommkclmclpchllfjekcdonpmejbdp*

Particulate matter measurement with the Raspberry Pi

Breathe deeply

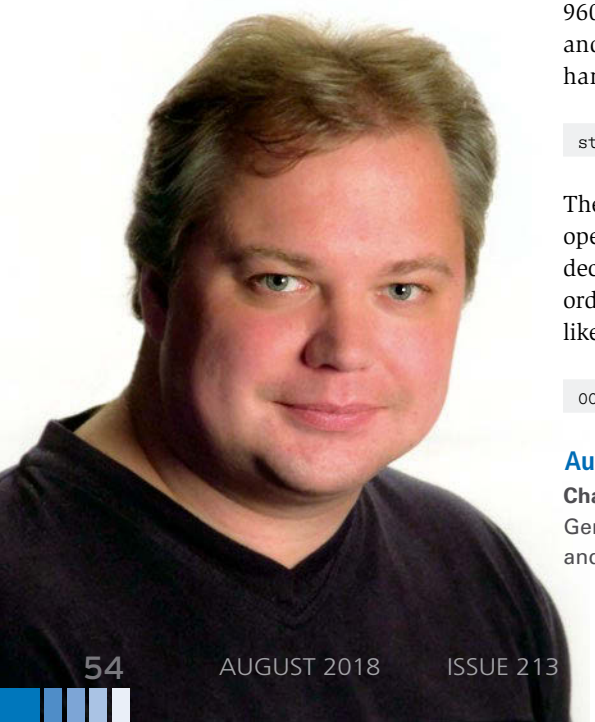
Whether you can breathe easily or not depends on what is in the air. Is there too much particulate matter that could be harmful to your health? A particulate matter measurement provides clarity – and the Raspberry Pi can help. *By Charly Kühnast*

How fresh is the air I breathe? To find out, I put my own particulate matter measuring station into operation. Particulate matter (PM) is a mixture of all kinds of organic and inorganic dusts, bacteria that travel through the air, and many other particles that do not exceed 10 micrometers in size.

The above measurement is why we use the term PM10. For comparison, a hair has a diameter of 50 to 75 micrometers. Today, even smaller particles with a size of only 2.5 micrometers (PM2.5) or smaller are considered separately because they penetrate deep into the airways and reach the alveoli.

There are several sensors on the market that can measure both PM classes. I chose the Nova SDS011 PM Sensor by Chinese manufacturer Nova Fitness Co., Ltd. The sensor draws air into a chamber and shoots a laser at it. It uses the dispersion of the reflected light to determine the particle size and quantity. It delivers the data via a serial interface.

A USB adapter is included with the sensor, making it easy to connect to the Raspberry Pi. After connecting, I can see from the syslog that the `/dev/ttyUSB0` interface is now available:



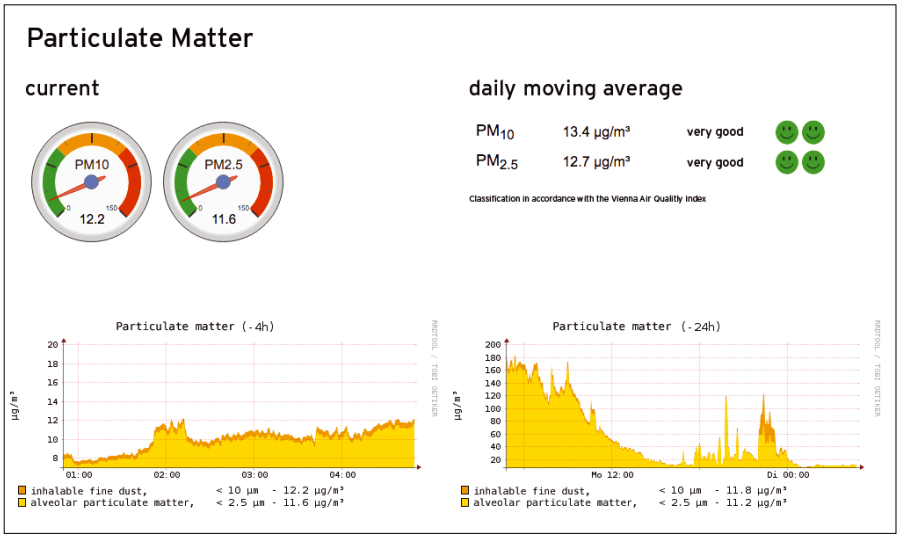


Figure 1: The dashboard for PM measurement.

```
[ 2.611448] usb 1-1.5: ch341-uart converter now attached to ttyUSB0
```

The manufacturer's data sheet [1] reveals that the interface must be set to 9600bps with 8 data bits, no parity, and one stop bit. The `stty` command handles this:

```
stty -F /dev/ttyUSB0 9600 raw
```

The data sheet also says that the sensor operator can expect the values in hexadecimal notation in a fixed, recurring order once per second. The output looks like this:

```
0000000 aac0 8c00 9400 bcf9 d5ab
```

This means that values are now available: `8c` for the PM2.5 class and `94` for PM10, 140, and 148 decimals.

The data sheet contains the following computation formula: The decimal value of the high byte is multiplied by 256, and the decimal value of the low byte is added to it. The result has to be divided by 10 – this is finally the PM value in micrograms per cubic meter of air. In the example, this would be 14 micrograms/m3 for the PM2.5 class, a low (i.e., good) value. You can then use RRDtool [2] to show your data in graph format (Figure 1). ■■■

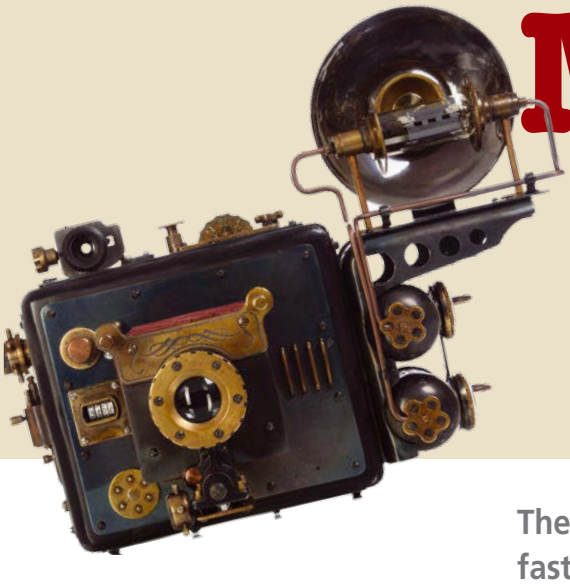
Info

[1] Laser dust sensor control protocol: *https://cdn.sparkfun.com/assets/parts/1/2/2/7/5/Laser_Dust_Sensor_Control_Protocol_V1.3.pdf*

[2] RRDtool: *https://oss.oetiker.ch/rrdtool/*

Author

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



MakerSpace

An alternative to the Raspberry Pi 3

ODROID-C2

The inexpensive ODROID-C2 module runs two to three times faster than a Raspberry Pi 3. *By Pete Metcalfe*

The ODROID series of single-board computers is manufactured by Hardkernel [1] out of South Korea. The ODROID-C1+ ($35) and the ODROID-C2 ($46) have a form factor similar to the Raspberry Pi 3 (RPi3) (Figure 1). The higher end ODROID-XU ($59), which is around five times faster than the RPi3, has a significantly different board layout.

I've been happy with the functionality and openness of the Raspberry Pi platform; however, I find its desktop performance a bit sluggish. For only a few dollars more than an RPi3, the ODROID-C2 CPU, RAM, and GPU specs are impressive (Table 1). My goal was to see whether I could use the ODROID-C2 for some typical Raspberry Pi applications.

First Impressions

The ODROID-C2 has almost the same footprint as the RPi3, but not exactly. Because the microSD mount is different, I could use some, but not all, of my Pi cases (Figure 2). When you are designing your projects, it is important to note that the ODROID-C2 does not have built-in WiFi or Bluetooth adapters, so you'll need wired connections or USB adapters. Like some of the Orange Pi modules, the ODROID-C2 has a built-in infrared (IR) connection.

ODROID-C2 can be loaded with Ubuntu, Arch Linux, or Android images. For testing, I used the Armbian 5.40 Ubuntu desktop, and the performance was significantly faster than my RPi3 Raspbian desktop. I could definitely see ODROID-C2 being used as a low-cost web browser station.

The ODROID-C2 images are quite lean, so you will need to go to the ODROID wiki [2] for instructions on loading additional software components.

The ODROID-C2 has 40 general-purpose input/output (GPIO) pins arranged as on the RPi3, so it is possible to use Pi prototyping HATs on the ODROID-C2 (Figure 3). Some differences in the pin definitions between the two modules are noticeable, so for this reason, I didn't risk using any of my intelligent Pi

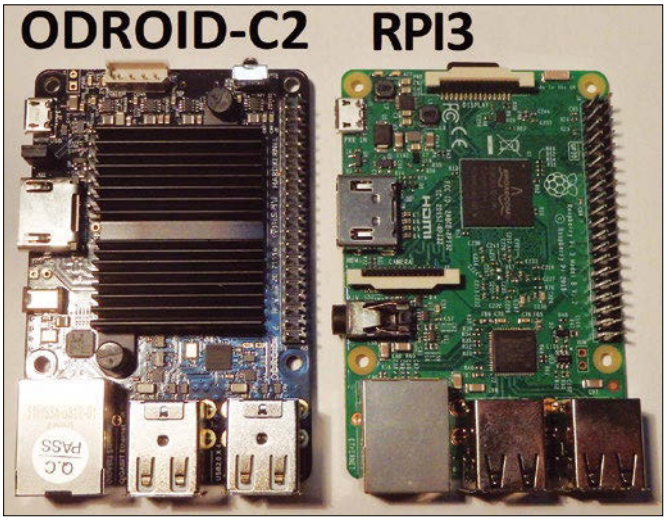


Figure 1: ODROID-C2 and RPi3 compared.

Table 1: ODROID-C2/RPi3 Hardware Comparison

Feature	ODROID-C2	Raspberry Pi 3
CPU	Amlogic S905 SoC, 4x ARM Cortex-A53 1.5GHz, 64-bit ARMv8 @28nm	Broadcom BCM2837, 4x ARM Cortex-A53 1.2GHz, 64-bit ARMv7 @40nm
RAM	2GB 32-bit DDR3 912MHz	1GB 32-bit LPDDR2 450MHz
GPU	3x ARM Mali-450 MP 700MHz	1x VideoCore IV 250MHz
USB Ports	4	4
Ethernet/LAN	10/100/1000Mbps	10/100Mbps
Built-in WiFi	No	Yes
Built-in Bluetooth	No	Yes
IR Receiver	Built-in	Needs add-on
I/O Expansion	40+7-pin port, GPIO/UART/I2C/I2S/ADC	40-pin port, GPIO/UART/SPI/I2C/I2S/
Camera Input	USB 720p	MIPI CSI 1080p
List Price (US)	$46	$35

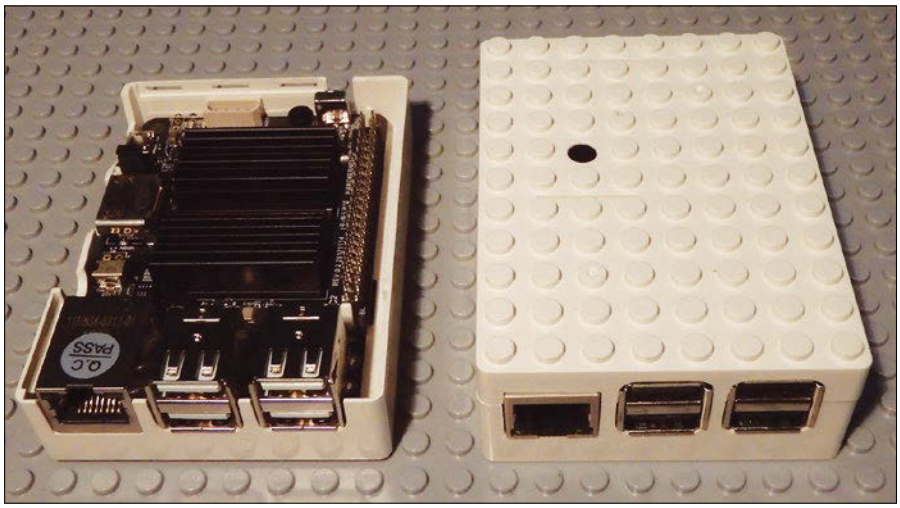


Figure 2: ODROID-C2 in a Pi Lego case.

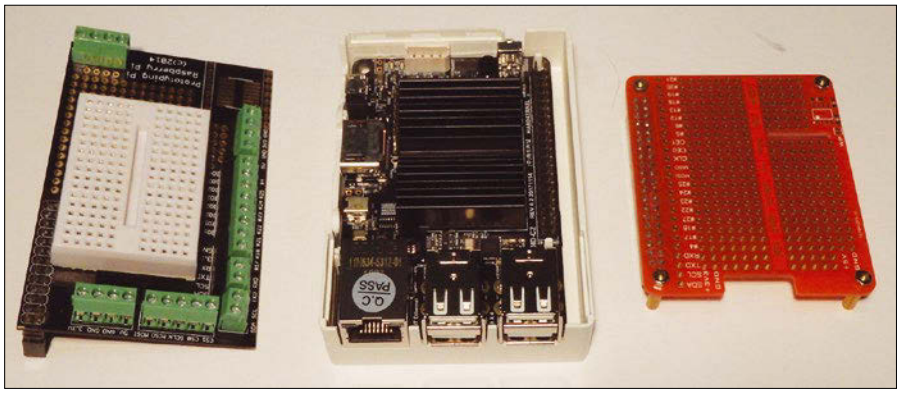


Figure 3: Pi prototyping HATs (left and right) with an ODROID-C2 (center).

Listing 1: Pin Definitions

```
odroidc2:~$ gpio readall
 +------+-----+----------+------+ Model  ODROID-C2 +------+----------+-----+------+
 | GPIO | wPi |   Name   | Mode | V | Physical | V | Mode |   Name   | wPi | GPIO |
 +------+-----+----------+------+---+----++----+---+------+----------+-----+------+
 |      |     |     3.3v |      |   |  1 || 2  |   |      | 5v       |     |      |
 |      |   8 |    SDA.1 |      |   |  3 || 4  |   |      | 5V       |     |      |
 |      |   9 |    SCL.1 |      |   |  5 || 6  |   |      | 0v       |     |      |
 |  249 |   7 | GPIO.249 |   IN | 1 |  7 || 8  |   |      | TxD1     | 15  |      |
 |      |     |       0v |      |   |  9 || 10 |   |      | RxD1     | 16  |      |
 |  247 |   0 | GPIO.247 |   IN | 1 | 11 || 12 | 1 | IN   | GPIO.238 | 1   |  238 |
 |  239 |   2 | GPIO.239 |   IN | 1 | 13 || 14 |   |      | 0v       |     |      |
 |  237 |   3 | GPIO.237 |   IN | 1 | 15 || 16 | 1 | IN   | GPIO.236 | 4   |  236 |
 |      |     |     3.3v |      |   | 17 || 18 | 1 | IN   | GPIO.233 | 5   |  233 |
 |  235 |  12 | GPIO.235 |  OUT | 0 | 19 || 20 |   |      | 0v       |     |      |
 |  232 |  13 | GPIO.232 |   IN | 1 | 21 || 22 | 1 | IN   | GPIO.231 | 6   |  231 |
 |  230 |  14 | GPIO.230 |  OUT | 0 | 23 || 24 | 1 | IN   | GPIO.229 | 10  |  229 |
 |      |     |       0v |      |   | 25 || 26 | 1 | IN   | GPIO.225 | 11  |  225 |
 |      |  30 |    SDA.2 |      |   | 27 || 28 |   |      | SCL.2    | 31  |      |
 |  228 |  21 | GPIO.228 |   IN | 1 | 29 || 30 |   |      | 0v       |     |      |
 |  219 |  22 | GPIO.219 |   IN | 1 | 31 || 32 | 1 | IN   | GPIO.224 | 26  |  224 |
 |  234 |  23 | GPIO.234 |   IN | 1 | 33 || 34 |   |      | 0v       |     |      |
 |  214 |  24 | GPIO.214 |   IN | 1 | 35 || 36 | 1 | IN   | GPIO.218 | 27  |  218 |
 |      |  25 |    AIN.1 |      |   | 37 || 38 |   |      | 1v8      | 28  |      |
 |      |     |       0v |      |   | 39 || 40 |   |      | AIN.0    | 29  |      |
 +------+-----+----------+------+---+----++----+---+------+----------+-----+------+
```

HATs on the ODROID-C2. The `gpio` command-line tool shows the pin definitions (Listing 1).

The Pi's GPIO numbers range from 2 to 27, whereas the ODROID-C2 pins are in the 200s; therefore, don't expect to be able to run all your Raspberry Pi code "as is" on the ODROID-C2.

Unlike Arduinos, the Raspberry Pi platform has no built-in support for analog inputs, so I got pretty excited when I noticed that the ODROID-C2 had two built-in Analog-to-Digital Converter (ADC) pins (AIN.1 on pin 37 and AIN.0 on pin 40). However, after some investigation, I found that these pins had virtually no example code, and they only support 1.8V. Most of my analog input sensors require 3.3V or 5V, so I'm not sure how often these ADC pins will be used.

Python Applications

The ODROID-C2 wiki references the RPi.GPIO and WiringPi Python libraries. I tested both of these libraries and found that standard reads and writes worked, but neither of these libraries supported the callback functions like the Raspberry Pi versions. For existing Pi projects in which you are using callback functions for rising and/or falling digital signals (e.g., intrusion alarms), you will need to recode with a polling method.

Also note that the ODROID RPi.GPIO library is a little confusing because it uses the Pi pin names and not the ODROID pin names. For example, ODROID-C2 physical pin 7 is referenced as GPIO.04 (as on a Pi) and not GPIO.249 (the ODROID-C2 name). Listing 2 is a simple Python example that polls for a button press and then toggles an LED output when a button press is caught.

Some excellent Python libraries are designed to work with the Raspberry Pi; however, it will require some trial and error to determine which libraries will and won't work with the ODROID-C2. I tried testing the DHT11 temperature and humidity sensor with the ODROID-C2, and unfortunately the library had a *Segmentation Error* when I tried running an example.

Node-RED

Node-RED can be installed on ODROID-C2 by using the manual install instructions for Raspbian at the Node-RED website [3]. This install procedure will add a start item to the desktop Application

Listing 2: Python Read/Write Example

```
01 #
02 import RPi.GPIO as GPIO
03
04 GPIO.setmode(GPIO.BCM)
05
06 button = 4  # physical pin 7, PI GPIO.04, ODROID-C2 GPIO.249
07 led = 17   # physical pin 11, PI GPIO.17, ODROID-C2 GPIO.247
08 GPIO.setup(led, GPIO.OUT, initial=GPIO.LOW)
09 GPIO.setup(button, GPIO.IN, pull_up_down = GPIO.PUD_UP)
10
11 print ('Wait for button...')
12 while True:
13         if GPIO.input(button) == 1:
14                 GPIO.output(led,0)
15         else:
16                 GPIO.output(led,1)
17                 print "button pressed"
```

Listing 3: led.c – Read/Write Example

```
01 #include <wiringPi.h>
02
03 int main(void)
04 {
05     wiringPiSetup();
06     int led = 0;
07     int button = 7;
08     pinMode(led, OUTPUT);
09     pinMode(button, INPUT);
10
11     for (;;)
12     {
13         if (digitalRead(button) == 1) {
14            digitalWrite(led, LOW);
15         }
16     else
17     {
18            digitalWrite(led, HIGH);
19         }
20     }
21     return 0;
22 }
```

menu, but because of hardware differences, the Raspberry Pi GPIO input/output components will not load. To read and write to the GPIO, a simple workaround is to use *exec* nodes to call the `gpio` utility (Figure 4). The command-line syntax for writing and reading with `gpio` is:

```
gpio write <pin state>
gpio read <pin>
```

One of the limitations of this workaround is that you will need to add a polling mechanism. Luckily, you can use a good scheduling node like *Big Timer*.

C Applications

Programming in C is fairly well documented, and an ODROID "C Tinkering Kit" is sold separately. The WiringPi library is used in C applications. Listing 3 is a C version of the Python code in Listing 2. It is important to note that these two examples talk to the same physical pins, but the C WiringPi library uses the ODROID-C2 *wPi* numbers, whereas the Python RPi.GPIO library uses the Pi *BCM* numbers.

To compile and run the program in Listing 3, enter

```
$ gcc -o led led.c -lwiringPi -lpthread
$ sudo ./led
```

in a terminal.

A Streaming Video Server Example

The ODROID-C2 does not have a camera serial interface (CSI) port like the Raspberry Pi, but you can use generic USB webcams. I built two identical video servers using Lego Pi cases and some added Lego blocks to secure the USB webcam and cables (Figure 5).

Some good video server software options are available. I like to use `motion`, because it is super-easy to set up. To install it, enter:

```
sudo apt-get install motion
```

Once `motion` was installed on both my RPi3 and ODROID-C2, my goal was to try and run both systems at 800x600 resolution with 24 frames/second (fps) refresh and see how the performance varied. For this setup, I edited some parameters in `/etc/motion/motion.conf` (Listing 4).

To run the video server, enter:

```
sudo motion
```

The `motion` package has a built-in web server that you access at *http://<your_ip>:8081* (Figure 6).

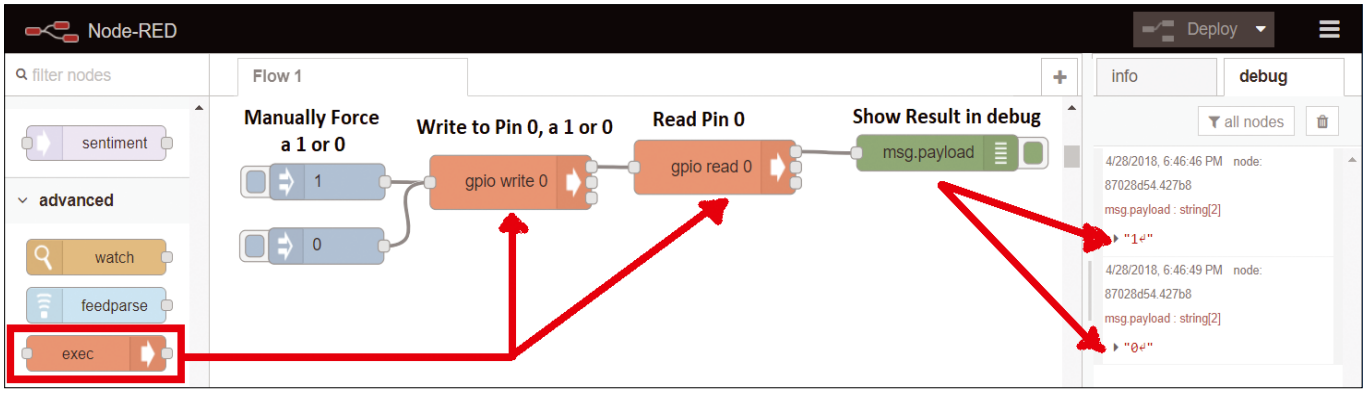


Figure 4: GPIO access in Node-RED.

Listing 4: /etc/motion/motion.conf

```
# Image width (pixels). Valid range: Camera-dependent, default: 352
width 800

# Image height (pixels). Valid range: Camera-dependent, default: 288
height 600

# Maximum number of frames to be captured per second.
framerate 24

# Maximum framerate for streams (default: 1)
stream_maxrate 24

# Restrict stream connections to localhost only (default: on)
stream_localhost off
```

Listing 5: Performance Check with vmstat

```
$ vmstat # Pi 3 video server at 24fps
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 1  0      0 593552  17808 269856    0    0    41    57  803  180  7  1 74 18  0

$ vmstat # ODROID-C2 video server at 24fps
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 1  0      0 863964  47136 463628    0    0     1     1 1161  286  2  1 97  0  0
```

The RPi3 had issues trying to stream 24fps, and the video would often freeze up. The faster specs of the ODROID-C2 allowed for near perfect real-time video streaming. To get a quick check on the differences in performance, I used the `vmstat` command-line tool on the RPi3 and ODROID-C2 (Listing 5).

The ODROID-C2 was running at 97% idle time (*id*) compared with the Pi's 74%. Probably one of the biggest performance differences was the time the CPU was waiting for I/O (*wa*, 18% for the PI compared with 0% for the ODROID-C2). The ODROID-C2 was the clear winner as a live-streaming video server.

Summary

Being able to reuse some of my Pi cases and prototyping HATs with the ODROID-C2 was nice. As a Pi user, I found that coding in C, Python, and Node-RED on the ODROID-C2 was fairly easy, although with many limitations compared with the Pi platform. The ODROID wiki had key product documentation, but it was nowhere near the incredibly rich documentation that exists with Raspberry Pi modules.

Some excellent Python libraries and Pi hardware add-ons that support a variety of sensors and I/O applications might or might not work with the ODROID hardware.

During the development cycle of a project, it is nice to have a fast interface, but typically my final projects do not need any high-end processing, and they can often run on Raspberry Pi 1 modules. For now, I would stick to a Raspberry Pi for GPIO/hardware projects.

I enjoyed playing with the ODROID-C2. For projects requiring higher performance, such as video servers and graphic or web applications, the ODROID-C2 module is definitely worth considering. ■■■

Info

[1] ODROID website: *http://www.hardkernel.com*

[2] ODROID wiki: *https://wiki.odroid.com*

[3] Node-RED: *https://nodered.org*

Author

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

Figure 5: RPi3/ODROID-C2 video server.

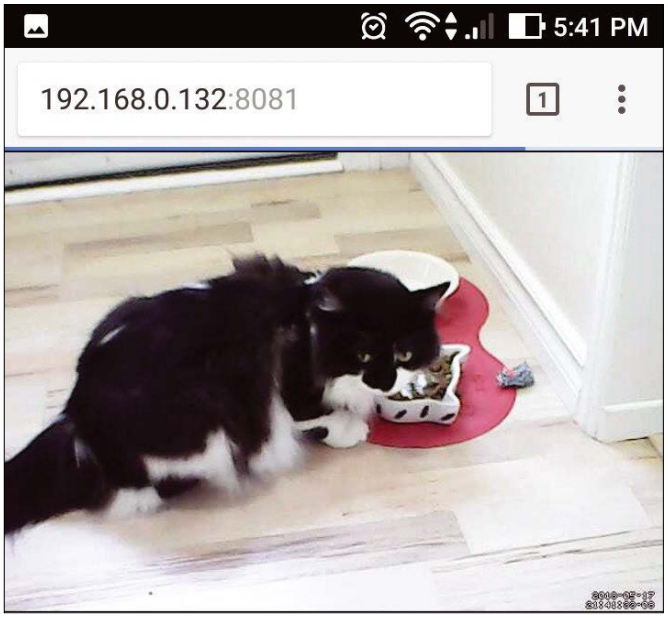


Figure 6: ODROID-C2 live video.

MakerSpace

Nine handpicked projects for the Raspberry Pi

Raw Diamonds

The Raspberry Pi success story has inspired thousands of projects. In this article, we present a selection of Raspberry Pi gems for hobbyists, researchers, and industrial use.

By Ferdinand Thommes

The Raspberry Pi originated with the idea of teaching children to program, and sales skyrocketed, with more than 18 million units sold to date. A new model with improved hardware, the Raspberry Pi 3 Model B+, was recently released. The applications range from baby monitors to the 300-node Raspberry Pi cluster [1] at the Free University of Bozen-Bolzano, Italy, and the 750-node HPC testbed [2] at Los Alamos National Laboratory, New Mexico. SUSE even supplies its Enterprise Server 12 for the single-board computer (SBC) [3]. In this article, I reveal my personal list of the best projects.

NextCloudPi

Hard-working developer Nacho Parker from the Nextcloud environment maintains a NextCloudPi (NCPi) image for the Raspberry Pi that connects the standard Raspbian distribution with the Nextcloud storage software (Figure 1). As he told *Linux Pro Magazine*, one other developer besides him works on the distribution code. The continuously updated image is based on Raspbian 9 and is adapted to the current Nextcloud version 13.0.1. NCPi [4] is available as an ARM image or x86 Docker container.

The code for NCPi is available from GitHub [5], where instructions for using the build script for the Raspbian image, an x86 image, and a Docker ARM hard-float (armhf) image are provided. In addition to the prebuilt Raspberry Pi image, Docker images for the Raspberry Pi, the x86 platform, and hardware based on armhf can be found there. The server scores up to 1,000 NCPi downloads per day.

NCPi includes its own setup tool and supports automatic security updates, Let's Encrypt, and Fail2Ban, as well as Redis, APCu PHP cache, and PHP Zend OPcache. Apps for calendars, contacts, tasks, and notes are preinstalled. Lately, NCPi supports the ZRAM kernel module, which serves as a swap or generic RAM disk.

If you are looking for an operating system that is always up to date and already contains a meaningfully preconfigured Nextcloud installation, you will certainly find it with NCPi. Support is provided by a specially set up Nextcloud subforum [6].

According to Parker, developers are currently focusing on improving the usability of the web interface, extending the functions for the Docker version, and building images for additional SBCs.

Pi-hole

The Pi-hole "DNS sinkhole" protects all the devices on your home network from banners and trackers. It is based on the

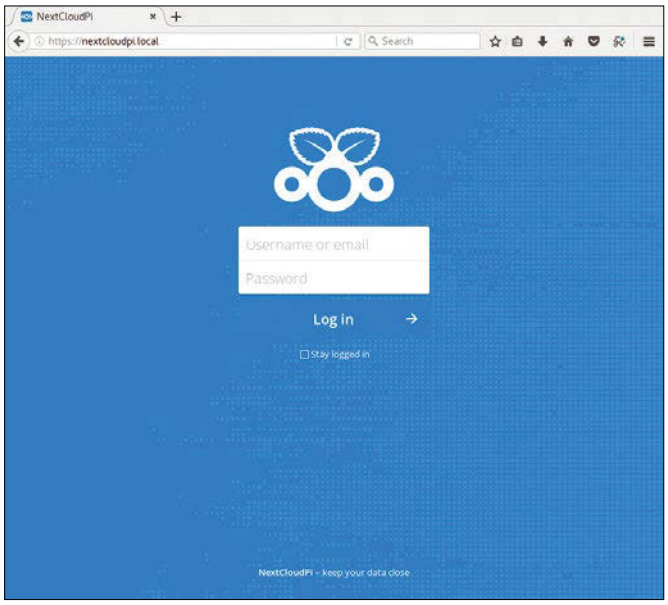


Figure 1: NextCloudPi connects the Raspberry Pi with the Nextcloud file-sharing software.

lightweight web servers Lighttpd and Dnsmasq. Jacob Salmela has been working on the project since 2015.

From a technical point of view, Pi-hole internally manages a list of approximately 120,000 domain names that deliver advertisements or trackers to users. Because Pi-hole acts as a DNS server on the local network, it effectively blocks unwanted domains. A website requested on the network cannot even load the advertisements placed on it.

Install Pi-hole on the Raspberry Pi with the command:

```
curl -sSL ➥
  https://install.pi-hole.net | bash
```

The configuration that follows only prompts the user for some basic data about the network. You can add further block lists through a web interface. Afterward, you need to specify the Raspberry Pi as the DNS server for your network. The easiest way to do this is via the router, for which you can find detailed instructions online [7].

Because Pi-hole blocks most advertising at the DNS level, no resources need to be turned off on the LAN. Pi-hole is particularly useful for in-app advertising on smartphones or apps on smart TVs. It can also serve as a DHCP server and is compatible with DNSCrypt. The project can be found on GitHub [8].

motionEyeOS

One common application for the Raspberry Pi is camera surveillance of rooms or outdoor areas. The Motion project [9] provides this service, be it to observe your pet or the hatching of the offspring in a bird's nest, or to monitor machine operations. Your Raspberry Pi can handle these tasks in combination with Motion. However, setting up such a solution is not trivial and requires experience in using the command line.

This is where motionEyeOS (Figure 2) comes in [10]. Under development since 2014, the system is based on BuildRoot [11] and uses Motion as the back end and motionEye as the front end.

The operating system can handle one or more cameras. Multiple Raspberry Pis can be interconnected via a hub or server and configured together. In addition to surveillance, an IP camera can serve as a webcam. motionEye can be used on a number of other SBCs or an x86 desktop under Linux. The wiki offers many detailed instructions [12].

According to the developer, Calin Crisan, the project has about 10,000 users. In the near future, he will be looking to provide an API for the project that users can use to build their own extensions and derived projects.

Speed Camera

Another application that integrates a camera comes from Canadian Claude Pageau. Speed Camera (Figure 3) is a Python program that uses the Open Source Computer Vision Library (OpenCV) to track the largest moving object in a camera's field of view and determine its speed and direction.

Speed Camera [13] records an image and optionally saves the data in a CSV file for later processing. If the traffic in front of the camera constantly exceeds the permitted speed, it can be recorded with this inexpensive solution. One field of application is sports – for example, to measure the speeds of runners or cyclists.

The software runs on Unix-based systems, Windows, and the Raspberry Pi and offers plugins for webcams as well as the Raspberry Pi camera module. A year ago, the project introduced the option of viewing stored CSV data and the corresponding images in a browser on the LAN. The developers have integrated a menu system, the `makehtml` converter, and a web server.

Figure 2: Adding a camera to motionEyeOS.

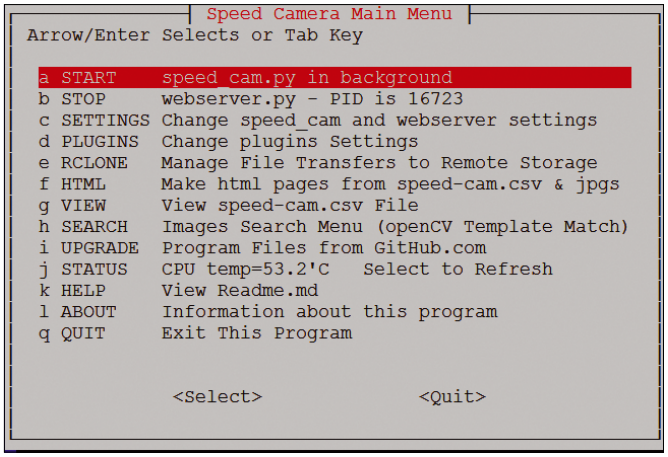


Figure 3: The Speed Camera dashboard.

Developer Pageau recommends at least a Raspberry Pi 2, because it has a quad-core CPU that performs better with threading.

Voice Kit

Google AIY stands for "do-it-yourself artificial intelligence." The project currently comprises two kits. Vision Kit uses a camera and the TensorFlow deep learning framework to detect objects, and the second AIY project kit discussed here, Voice Kit [14], supports natural speech recognition.

Anyone interested in doing so can turn a Raspberry Pi 3 (RPi3) into a Google Assistant in less than an hour. Beyond what the Google Assistant SDK already does, custom question-and-answer pairs are also possible. All of this fits in a handy cardboard cube containing a Raspberry Pi and a loudspeaker to help the Pi talk.

Besides an RPi3 and an SD card with a Voice Kit SD image [15], you need the Voice Kit itself ($10-$50, EUR27) [16]. In addition to the box, a speaker, and all necessary connection cables, the kit contains two expansion boards, the Voice HAT, and the Voice HAT microphone. The assembly procedure is well documented [17] and requires only a screwdriver. For makers and hackers, the maker's guide [18] provides source code and API references, as well as instructions for integrating additional sensors.

Cluster with Docker Swarm

Usually one Raspberry Pi is enough to realize a project, but some scenarios require concentrated computing power, where it makes sense to use a Pi cluster.

The described project [19] will work with just four Pis (Figure 4), with Docker Swarm [20] to manage them. Of course, far larger clusters can be built. Container specialist Resin built a Pi cluster known as The Beast with 144 boards and is currently working on an even larger model [21]. The cluster at the Free University of Bozen-Bolzano has no fewer than 300 Raspberry Pis.

Docker Swarm is not the only option for running Pi clusters. Kubernetes or Mesos Marathon are also suitable for this task. Unlike other solutions, though, Docker Swarm is already in place after installing Docker.

Initially, handling a small project is easier and the more sensible option for hobbyists. Joseph Tyler Jones, the developer of the howchoo maker platform, described the project. The simplest way to build a small tower with four RPi3 boards is to connect boards with inexpensive spacers in M2.5 format.

Jones used an ordinary Debian-based Raspbian operating system, to which you then need to add Docker. You then add the four boards to a Docker swarm via SSH, with the first node acting as the manager at the same time.

The first service Jones implements graphically displays the four nodes and tells you which containers they host. Additional services packed in containers can then easily be added to the setup and managed.

Raspberry Pi clusters are suitable not just for hobbyists and universities but also have applications in industry – especially in the field of metrology and control.

Revolution Pi

The Revolution Pi (RevPi) was designed by Kunbus GmbH [22] as an open source small controller for industrial use. Extensions also make it suitable as an Internet of Things (IoT) gateway or as an edge computer. It is designed as a modularly expandable DIN rail industrial PC housing and uses the Raspberry Pi Compute Module 3. Up to 10 expansion modules can be connected to a base module containing the CPU. Kunbus offers a choice of digital and analog I/O and gateway modules.

The RevPi (Figure 5) meets industry standard EN/IEC 61131-2, which defines requirements for programmable logic controllers (PLCs). In terms of software, Kunbus optionally offers various applications from several manufacturers. Logi.cals, for example, created the logi.CAD 3 development environment for creating PLC programs according to the IEC 61131-3 standard for execution on the RevPi [23].

RevPi is also interesting for home automation; the simplest version costs about EUR120 ($139).

When asked, Marketing Director Ekkehard Krebs said that open source was the key to the RevPi concept, so they could offer customers freedom of design and avoid creating unnecessary hurdles. The company has 30 developers, of which about 10 are working on the hardware, with customers in the four digits.

The consumer share is five percent. A new model, the RevPi Connect, was to join the six existing models in May, but production problems delayed delivery to mid-year. It will extend RevPi Core 3 by two RJ45 sockets and an RS485 connector, allowing it to be used as a gateway.

Figure 4: Raspberry Pis installed in a cluster. © howchoo.com

Figure 5: Revolution Pi: A Raspberry Pi with an industrial look. © KUNBUS GmbH

netPI

netPI was custom designed in a collaboration between Hilscher, a German communications and automation company, and the Raspberry Pi manufacturer Farnell. The device connects the circuit diagram of the RPi3, with the permission of the Raspberry Pi Foundation, to the netX industrial network controller chip from Hilscher to create an Open Edge Connectivity Ecosystem. A standard Raspberry Pi board is therefore not used.

netPI [24] is suitable for IoT and Industry 4.0 Edge Automation projects. It offers an expansion slot for network modules, and as a gateway, it integrates into shop floor operational technology networks such as Profinet, Ethernet/IP, and EtherCAT as an I/O slave and exchanges data bidirectionally with the cloud and IoT.

A Linux environment and Docker container virtualization enable customers to run their own developments or third-party software in isolated containers. For full compatibility, Docker applications can be developed on an RPi3 and then transferred by the programmers to the netPI. To familiarize customers with the possibilities of Docker, Hilscher offers several sample containers online [25]. The netPI software architecture complies with the IEC 62443 IT security standard.

When I inquired, the company explained that after an introductory period of just six months, they already have well over 100 customers. In addition to the development of further modules (e.g., for the wireless sector), Hilscher recently released the netPI CORE 3, in addition to the netPI RTE 3, available for EUR325 ($377). The CORE 3 version does without the industrial network controller and comes at a cheaper price of EUR225 ($261).

Janz Tec

Janz Tec offers two devices based on the RPi3 with the less than intuitive names emPC-A/RPI3 and emVIEW-7/RPI3. According to the company, Raspberry Pi-based devices are particularly popular for acquiring data measurements.

The idea for the 2015 RPi2B-based emPC-A/RPI was born after the release of a Codesys software variant for the Raspberry Pi [26]. Because many Janz Tec customers use this software, the Raspberry Pi was only missing a couple of interfaces and a robust housing to implement small industrial applications. The interfaces include a Controller Area Network (CAN) bus [27] developed by Bosch in 1986.

The newer RPi3 model additionally offers Bluetooth and wireless, thus enabling a complete industry-capable system. The emVIEW-7/RPI3 adds a Raspberry Pi touchscreen display to the emPC-A/RPI3.

According to the company, future developments will initially focus on the new RPi3 Model B+, which, in addition to a more powerful CPU, also offers more network throughput for Ethernet and wireless. ■■■

Info

[1] Giant Raspberry Pi cluster: *https://arxiv.org/pdf/1709.06815.pdf*

[2] Raspberry Pi HPC testbed: *http://www.lanl.gov/discover/news-release-archive/2017/November/1113-raspberry-pi.php*

[3] SLES for ARM: *https://www.suse.com/de-de/products/arm/raspberry-pi/*

[4] NCPi: *https://ownyourbits.com/2017/02/13/nextcloud-ready-raspberry-pi-image/*

[5] NCPi code: *https://github.com/nextcloud/nextcloudpi*

[6] NCPi support: *https://help.nextcloud.com/c/support/appliances-docker-snappy-vm*

[7] Pi-Hole guide: *https://discourse.pi-hole.net/t/how-do-i-configure-my-devices-to-use-pi-hole-as-their-dns-server/245*

[8] Pi-hole code: *https://github.com/pi-hole/pi-hole*

[9] Motion: *https://motion-project.github.io*

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

[11] BuildRoot: *http://buildroot.uclibc.org*

[12] motionEye wiki: *https://github.com/ccrisan/motioneyeos/wiki*

[13] Speed Camera: *https://github.com/pageauc/speed-camera*

[14] Voice Kit: *https://aiyprojects.withgoogle.com/voice/*

[15] Voice Kit SD image: *https://dl.google.com/dl/aiyprojects/aiyprojects-latest.img.xz*

[16] Voice Kit (the kit itself): *https://exp-tech.de/plattformen/raspberry-pi/8357/google-aiy-voice-kit-for-raspberry-pi*

[17] Voice Kit build instructions: *https://aiyprojects.withgoogle.com/voice/*

[18] Voice Kit maker's guide: *https://aiyprojects.withgoogle.com/voice/#makers-guide*

[19] Pi cluster: *https://howchoo.com/g/njy4zdm3mwy/how-to-run-a-raspberry-pi-cluster-with-docker-swarm*

[20] Docker Swarm: *https://docs.docker.com/engine/swarm/*

[21] Resin cluster: *https://resin.io/blog/the-evolution-of-the-beast-continues/*

[22] Revolution Pi: *https://revolution.kunbus.com/revpi-core/*

[23] logi.CAD 3: *http://help.logicals.com/display/LC3UserDocuEN/Putting+Raspberry+Pi+into+operation*

[24] netPI: *https://www.netiot.com/netpi/industrial-raspberry-pi-3/*

[25] Sample containers: *https://hub.docker.com/u/hilschernetpi/*

[26] Codesys: *https://www.codesys.com*

[27] CAN: *https://en.wikipedia.org/wiki/Controller_Area_Network*

MakerSpace

A chip to enable open hardware RISC-V

The new RISC-V chip promises to be a game changer in the open hardware field. *By Bruce Byfield*

The lack of open source computer chips has always been a major obstacle to the development of free hardware. The supply is limited, which can limit the number of open hardware units, and suppliers have been known to substitute proprietary chips without informing the manufacturer. In recent years, such problems have been alleviated by single-board microcontrollers, like those made by Arduino, but these devices are useful mainly for dedicated hardware with limited capacity. Now, however, the situation is on the verge of changing – in no small part because of the emergence of the RISC-V (pronounced "RISK Five") chip, which is starting to be used in proprietary and open hardware alike [1].

The idea of reduced instruction set computers, or RISC chips, has been around for several decades [2]. RISC refers to chips that have a greatly simplified set of instructions, which can make operations more efficient and reduce power expenditure. For example, rather than access to memory being included in most of the instructions that the chip uses, in a RISC chip, access to memory may be included only for operations in which it is needed. Examples of RISC-type architecture includes the PowerPC, MIPS, and SPARC, as well as the use of ARM in iOS and Android devices.

RISC-V is one of the latest implementations of the RISC concept. It originated in a three-month project in the summer of 2010 at the University of California, Berkeley, in their Electrical Engineering and Computer Sciences department's computer science division. Further development of the RISC-V architecture was funded by the Defense Advanced Research Projects Agency (DARPA), led by Krste Asanovic, Yunsup Lee, and Andrew Waterman. According to Jack Kang, the vice president of product and business development at SiFive [3], a company founded by the original developers, "the team was able to prove that smaller teams could design state-of-the-art silicon." The initial specifications were released under a BSD license, making the specifications an open standard that could be used for both proprietary and open purposes – a decision that played a major role in RISC-V's development, especially since no other chip specification was so accessible to users. The interest was immediate, and in 2015, the RISC-V Foundation was established: a non-profit organization that develops and promotes the RISC-V Instruction Set Architecture (ISA).

In the last three years, interest in RISC-V has been accelerating. Rick O'Connor, the executive director of the RISC-V Foundation describes RISC-V as attracting interest in research, academia,

and industry alike. "The Foundation's membership exceeded 100 organizations in November 2017," O'Connor says. "That's a 120 percent membership growth since November 2016, featuring both established technology giants and emerging startups."

In fact, the interest in RISC-V is worldwide. The latest in the ongoing series of RISC-V Workshops was held in Barcelona in May 2018, and another is scheduled in Chennai, India, in July 2018. Additionally, talks about RISC-V are scheduled later this year for the IEEE International Symposium on High Performance Computing and the Design Automation Conference in San Francisco. Recently, too, SiFive raised $50.6 million in startup capital [4]. Clearly, RISC-V is a standard whose time is about to arrive.

The Advantages of RISC

Why is RISC-V of such interest? To start, as O'Connor notes, chip development has reached a state where the existing architectures can do little to increase performance. "New heterogeneous, purpose-built accelerators are required to deliver ever-increasing performance needs." To all appearances, RISC-V appears to deliver the new approach that is required for the chip industry to continue development.

In the RISC-V State of the Union in Barcelona in May 2018, Krste Asanovic summarized some of the advantages that RISC-V delivers [5]. First, as the name implies, RISC-V is a simpler and smaller instruction set than other existing ISAs. It features what Asanovic describes as a clean state design that clearly separates user and root privileges. Even more importantly, RISC-V's memory model is unique in the way that it takes different types of users into account. Early on, RISC-V made the decision to keep the core as stable as possible, whenever possible adding new features through extensions. As O'Connor says, the "ISA base and standard extensions are frozen and are not expected to change in any way except for clarifications and improvements in its documentation." The result is an architecture that is unique among ISAs.

This decision shows in the memory model, which includes sections for the ISA core, with another section reserved for future developments. Vendors can make use of the reserved section, but at the possible expense of breaking compatibility with future versions of the ISA – a hazard that may be acceptable for a one-off device or one with a short shelf life. However, vendors are encourage to place any proprietary code into a custom section of the memory. As O'Connor points out, this arrangement allows for "user extensibility without breaking existing sections or incurring software fragmentation" (Figure 1).

However, the licensing is of equal importance. Besides the BSD licensing of the original code, the RISC-V Foundation is also developing a permissive license suitable for the formal specifications. No doubt free software advocates would prefer a copyleft license like the GNU General Public License (GPL), which would require all derivative works to use a similar license. The trouble is, as chip designer Andreas Olofsson points out,

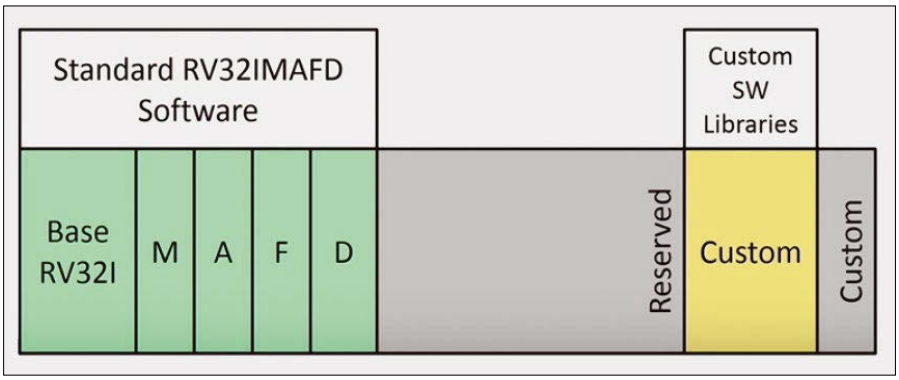


Figure 1: The RISC-V memory model includes reserved space for free-licensed core components and future common developments and custom space for proprietary extensions, which accommodates different types of users.

"The semiconductor industry would not take kindly to a copyleft license like GPL, which would force a company to open up everything inside the chip. Permissive licenses like the MIT license and BSD enable large risk-averse companies (with legal departments) to participate. With permissive licenses, there is always the risk that big companies will only take (and not give back), but with GPL style licensing in [use] they just won't get involved" [6].

As things are, RISC-V in effect acts as an open standard rather than a formal license. "RISC-V is license-free and royalty-free," O'Connor says, "which means you can build custom processors with zero licensing cost. Our intent is to provide a long-lived open ISA with significant infrastructure support." This includes documentation, compiler tool chains, operating system ports, reference software simulators, architectural test suites, and teaching materials." Although proprietary companies can use these tools for their short-term advantage in bringing products to market, small open hardware manufacturers gain access to what until now has been difficult or impossible to obtain. While not ideal by free software standards, the trade-off is perhaps acceptable. As Asanovic says, for all users, RISC-V offers "minimum wasted work through maximum reuses" of resources.

Risk Management

If, as seems likely, RISC-V adoption continues at the current rate, for the first time, hobbyists and small chip manufacturers will have a level playing field with the giants in the industry. A development board like HiFive (Figure 2), which was recently funded on Crowd Supply [7], will be working with chips similar to those of Nvidia graphics cards. Nvidia's proprietary Falcon microcontrollers, which have been used for more than a decade, will be replaced in the next generation of products with RISC-V [8]. Similarly, Western Digital is partnering with SiFive to produce an estimated billion processing cores [9]. Not only is such an alliance of an industry giant with a small newcomer almost unheard of, but the result will be hardware that should be much easier for open hardware manufacturers to produce.

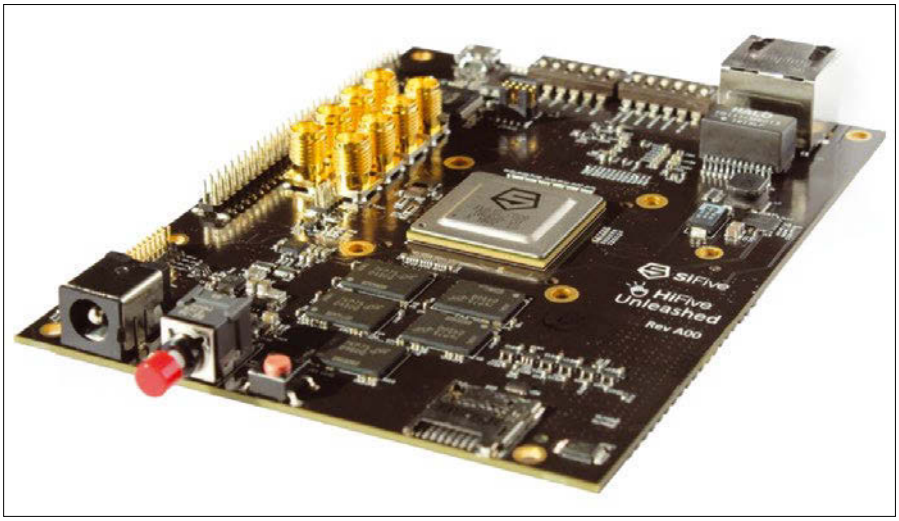


Figure 2: The HiFive development board is among the first to use a RISC-V chip.

Of course, some barriers will remain for open hardware. The custom space in the memory of RISC-V products will no doubt include proprietary code that must be reverse-engineered or replaced. Just as importantly, larger, long-established vendors will almost certainly have priority with manufacturers, forcing open hardware newcomers to wait at the back of the line behind those with high-volume orders. Still, if the RISC-V becomes as widely used as predicted, then many of the technical challenges will be removed. If that happens, the requirements for open hardware may shift from technical challenges to marketing and business development. In every sense of the word, RISC-V will have become a real game changer. ■■■

Info

[1] RISC-V Foundation: *https://riscv.org/*

[2] RISC history: *https://en.wikipedia.org/wiki/Reduced_instruction_set_computer*

[3] SiFive: *https://www.sifive.com/*

[4] SiFive startup capital: *http://bit-tech.net/news/tech/cpus/intel-capital-invests-in-risc-v-start-up-sifive/1/*

[5] RISC-V State of the Union: *https://www.youtube.com/channel/UC5gLmcFuvdGbajs4VL-WU3g*

[6] RISC-V licensing: *www.adapteva.com/andreas-blog/why-i-will-be-using-the-risc-v-in-my-next-chip/*

[7] HiFive: *https://www.crowdsupply.com/sifive/hifive-unleashed*

[8] Nvidia and RISC-V: *https://www.phoronix.com/scan.php?page=news_item&px=NVIDIA-RISC-V-Next-Gen-Falcon*

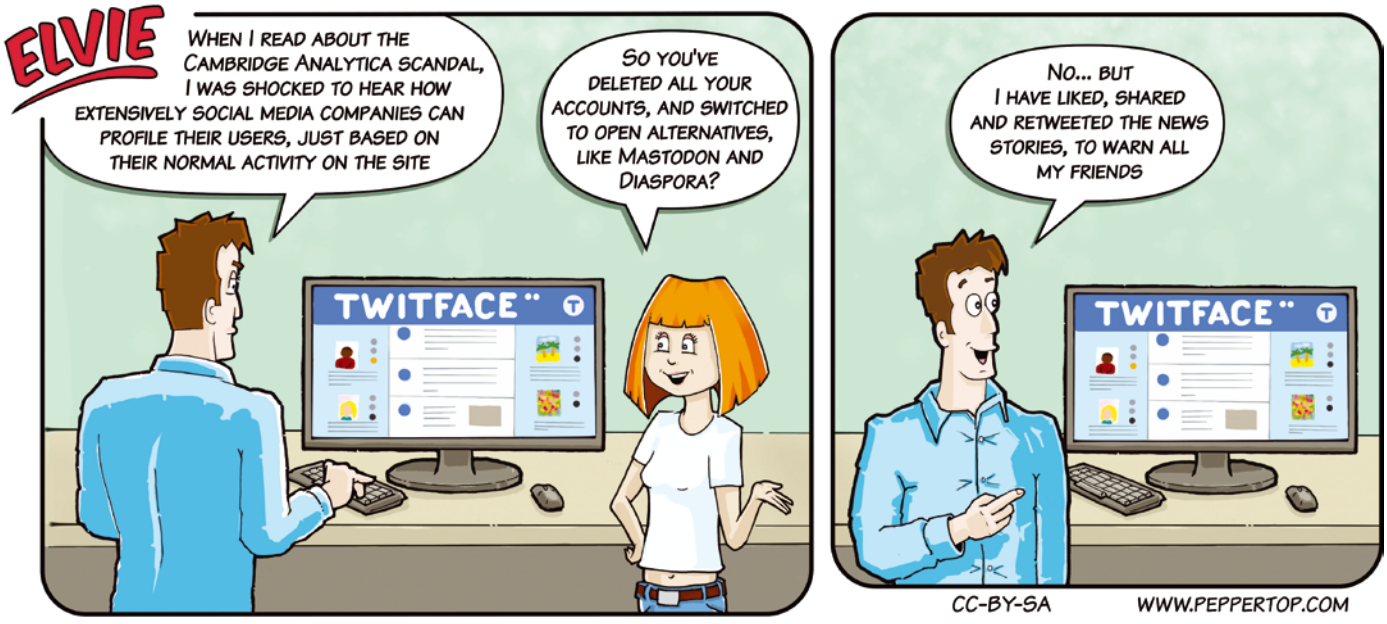
[9] Western Digital and RISC-V: *https://www.forbes.com/sites/tiriasresearch/2017/12/06/western-digital-gives-a-billion-unit-boost-to-open-source-risc-v-cpu/#11e2dcfa2266*

The amazing thing about Linux is that it can always take you one step more. You can already play music and edit videos using the powerful tools included free of charge on any standard Linux system, so you don't need a Linux magazine to tell you how to play an MP3 file. But we'd rather help you with the next step. Like, for instance, suppose you want to set up a music server on your home network, so you can play music from anywhere in your house and control it from a notebook or smartphone? This month we look at the Music Player Daemon (MPD) and the Cantata front end – a cool tool combination that could forever change the way you play your music.

Also inside, Paul Brown continues his series on video editing tools with a look at Kdenlive and ImageMagick, and we show you how to impose order on your recipe collection with Gnome Recipes.

Image © Olexandr Moroz, 123RF.com

LINUXVOICE



MADDOG'S DOGHOUSE

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®.

While Free and Open Source Software and Hardware is becoming more commonplace, the battle against closed source isn't over. BY JON "MADDOG" HALL

The fight for FOSSH continues

A couple of months ago, I wrote about the significance of the year 2019, how it marked several significant anniversaries, and how we might start planning to bring forward these milestones to the general public.

This started me thinking about the progress that GNU/Linux has made over the past quarter-century and how Free and Open Source Software and Hardware (FOSSH), in general, helps to drive the world.

The fact that FOSSH is so prevalent in the world might encourage us to relax a bit and declare "victory." After all, even Microsoft says they love open source.

However, I think that the playing field of "open" versus "closed" may only be moving higher up the food chain, with many people using and supporting FOSSH at the middle and lower levels, while higher level functionality is still closed, even though companies may be using FOSSH to help develop their products.

Artificial Intelligence (AI) and Internet of Things (IoT) are examples of the "higher level" of applications that may or may not be open, and both of these could have rather frightening implications if not handled properly. Many articles have been written about the issue of having IoT "things" get infected with viruses and (as an example) coordinate a denial of service attack on the Internet.

Closed source might prevent the people responsible for deploying those "things" from getting a security patch in a timely manner if the company that made the "thing" is no longer in business.

An even scarier thought is if the "thing" infected by the virus is artificially intelligent. This has been the fabric of many science fiction stories.

We (all of us) have to keep pushing the concepts and benefits of openness. We have to keep educating others on the business models around FOSSH and shooting down the (sometimes innocently spread) FUD (fear, uncertainty, and doubt) that fights against openness.

It pains me that after all this time there are still companies that produce hardware products that use closed firmware (you know the companies I am talking about) and who refuse to document how the hardware works so that the FOSS community can properly program that hardware.

I often think how fast these companies would change their strategy if all of their customers simply said "we will not buy your hardware unless your interfaces are published and your hardware is able to be programmed with FOSS."

This does not mean that the companies cannot also generate their own closed firmware if they feel that they can create an advantage over their competitors, but it does mean that certain customers could now buy their hardware knowing that FOSS could support it properly.

Years ago, Digital Equipment Corporation (DEC) had a math library for Unix that was one of the best in the world. DEC paid a mathematician money to create an efficient library of routines like `sin()`, `cos()`, etc., and refused to disclose the source code, saying their competitors would easily be able to duplicate their work and they would lose their advantage.

While DEC was willing to donate the binary library to the Alphalinux project, they refused to donate the sources.

The Alphalinux community wanted the shipping math library to have sources available, so routine by routine they rewrote the subroutines until all of the subroutines except one were faster than the DEC binary-only version. That last subroutine was "fast enough," so no one spent the time to make it any faster.

We have come so far in FOSSH, but we need to push harder and not let companies designate FOSSH as a tool to help them create closed systems at some other level.

We have to talk to the consumers of tech and get them to understand and care about having FOSSH products.

Recently I was advising a major government funder of research and business about the benefit of using FOSSH in creating new products and industry. While they knew about FOSSH and its benefits for an economy to a certain level, two hours of a rapid machine-gun presentation brought the concepts they had in the back of their minds up to the frontal lobes. I advised that companies requesting public money to do research and incubate companies (public or private) owe to the public a commitment to FOSSH, and public entities owe it to their constituents to require FOSSH as a basis for products and services purchased.

Even fully private concerns with private funding should consider using, specifying, buying, and producing FOSSH products and services, because the opposite of freedom is slavery, and software slavery means loss of control over your business. Business people love control, and having them recognize their need for control over how and when they deliver products and services can drive them to FOSSH when discussions of freedom have no effect. ■■■

openSUSE Conference 2018

The openSUSE community assembles in Prague. BY SWAPNIL BHARTIYA

Prague is like the second headquarters of the openSUSE community. After Nuremberg, which is home to SUSE's headquarters, Prague has the largest openSUSE developer base. No surprise that this year's openSUSE conference was organized in the capital of Czech Republic. OpenSUSE PR manager Douglas DeMaio told me last year that he secretly wanted the next conference in Prague so he could get a break from organizing it. But when I landed in Prague, I saw him running around, sweating in the cold weather of the region, helping the community. Once you are part of a friendly open source community, there are no excuses: Everyone pitches in.

I had been at the OpenStack Summit in Vancouver, Canada, and I cut my trip short to fly directly to Prague to cover the conference and meet my friends from the openSUSE community. I was extremely jet lagged, but the moment I stepped into the venue, I found myself reinvigorated as I greeted friends I have known for many years.

Highlights of the Show

The biggest highlight of the show was the release of openSUSE Leap 15. The Leap release manage, Ludwig Nussel, hit the green button amid some fanfare to officially announce the release. The second biggest story that came out of the conference was the forking of Red Hat Spacewalk, a systems management solution that serves as the base for SUSE Manager and Red Hat Satellite. In an interview, Klaus Kämpf, SUSE's project owner for SUSE Manager, told us that Red Hat put the project in maintenance mode and did not have enough developer resources to accept the changes SUSE developers were submitting. SUSE tried to work with Red Hat to take over the project so they could maintain it, but that didn't work out. Since Spacewalk is the foundation of SUSE Manager, the only way for SUSE to maintain the project was to fork it. The new project is called Uyuni.

Pi for Health

OpenSUSE was one of the first Linux-based distributions to support the latest 64-bit Raspberry Pi boards. The community is known for giving away Pis to the GNU Health project. Andrew Wafaa, who gets paid by ARM to work on openSUSE, handed over a box of 10 Raspberry Pis to the project's founder Luis Falcon.

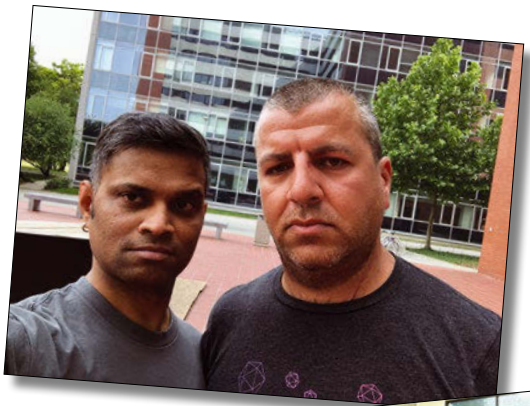
I got many insider stories and scoops at this year's conference, some of which you may read about in coming issues of this magazine. And I got word of some big changes that are coming to openSUSE. SUSE's Richard Brown gave me a sneak peak of what to expect from the next release of openSUSE and probably SUSE Enterprise Linux. The SUSE team has been working on borrowing some ideas from Kubic (a platform for containerized workloads) and bringing transactional updates to openSUSE.

Beer Time

What kind of German company or community would host an event without any beer? You often get beer after the event, but Ana María Martínez Gómez, of the openSUSE Build Service, went full Bavarian and introduced a new track called "lightning beer talks." The format was simple: The presenter has to hold a beer, a microphone, and a projector remote in their hands and deliver a talk while drinking beer. Wafaa didn't deliver a talk, but he drank the beer and gave us a dance that was well appreciated.. *(https://www.youtube.com/watch?v=EM3h4bAzDI4)*

On the final day of the event, an annual board meeting presented a new openSUSE board to the community. A drawing during the meeting gave away a brand new TUXEDO InfinityBook Pro 13 laptop, which runs on openSUSE.

A party was planned for after the meeting, but DeMaio and I took a long walk to enjoy some spicy Indian food and Czech beer. I was back to DC the next day, getting ready for my next open source conference. ■■■



Online Jukebox

The MPD server helps you manage your music collection flexibly and remotely and integrates external content.

BY ERIK BÄRWALDT

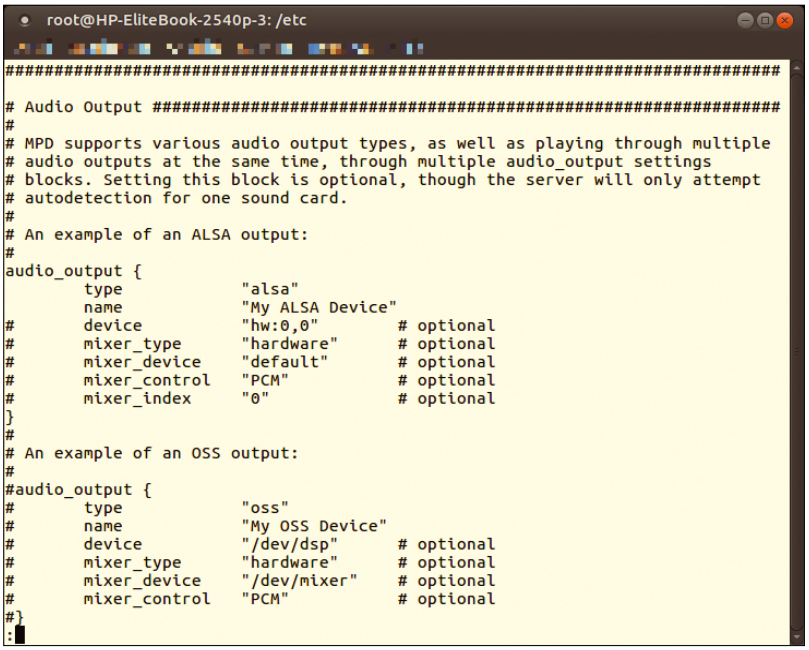
For a long time, Linux had a reputation as not being suitable for multimedia applications because of a lack of appropriate codecs and applications. The tide has turned, and the package sources of most distributions contain more multimedia programs than many other operating systems. Almost every standard installation comes with audio players that play music files in all common formats. These programs usually only run locally on a desktop computer, where the audio files are also located. The players cannot be used on the web – or only to a very limited extent.

The Music Player Daemon (MPD) [1] lets users enjoy music far more flexibly and from any location: It acts as a music server on a standalone system or on the local network completely independent of the clients. On standalone systems, the advantages of MPD are less obvious than on the network, because the server and client run on the same machine and the function is thus similar to any conventional playback software for audio files. On your home network, for example, you can connect an MPD server to a hi-fi system and then control it with a notebook or smartphone. The streaming capabilities of the MPD server also let you play audio files on remote clients, if desired.

The MPD server can be found in the package sources of almost all major distributions, from Debian and Fedora to Mageia and from openSUSE to Slackware and Ubuntu. It can therefore be installed with just a few mouse clicks. However, configuring the service requires some manual work: As an administrative user, you will be concerned with the `/etc/mpd.conf` file, which you can open in an editor. This file groups all the important server settings.

Because MPD manages the audio files in a database, you need to change the path in the `music_directory` section in `mpd.conf` for where you want the service to search for music files. MPD can handle numerous file formats, from classic MP3s to the free Ogg Vorbis and FLAC formats. The software also stores individual playlists in a separate path on the server. Enter this in the `playlist_directory` section of the configuration file.

Figure 1: The MPD server configuration takes place in a single file.



Architectural

Linux comes with several historically grown sound architectures, but not all of them support all applications. The MPD server shows its flexible side: It supports the most common sound architectures with OSS or ALSA and works in combination with PulseAudio. Furthermore, it gets on with the JACK audio server, which is more established in the professional sector.

Which sound architecture is used depends on the operating system and the hardware. You need to make the appropriate settings in the `audio_output` section of the server configuration file, where you will find preconfigured entries for the (default) ALSA subsystem and the older OSS sound architecture. You can enable them by deleting the hash sign (#) at the beginning of the corresponding lines. Conversely, you can disable entries by placing a hash sign at the start of the respective line (Figure 1).

After changing the basic settings, launch the MPD server with:

```
sudo service mpd start
```

If the server was already active when the system rebooted, you can stop and reactivate it with the new settings by typing:

```
sudo service mpd restart
```

The server now waits for connection requests from the clients on the local network, default port 6600.

Clients

The many Linux MPD clients have different desktop-specific interfaces and ranges of functions. Cantata [2] is a relatively new but functionally mature front end based on the Qt5 libraries, although it also works under other graphics libraries.

Most popular distributions have Cantata in their repositories, so the program usually can be installed with a click of the mouse. The installation routine creates a starter in the Multimedia or Entertainment Media submenu. The program opens a clear-cut wizard for the configuration steps after the first start.

The first dialog gives some hints about the MPD server. In the second dialog, you define the operating mode: The function is available as a front end for an MPD server on the network or for use on the local system. For use as a client on the network, specify the IP address of the computer on which the MPD daemon is running (also see the "Static IP" box). If this requires authentication, enter the corresponding password in the dialog. You do not usually need to change the port number (Figure 2).

You additionally need to define the directory where the music files are located. If the MPD server is running on the same system as Cantata, just enter *localhost* in the Host input field. Once the configuration is complete, click the *Connect* button to establish the connection between Cantata and the MPD server.

In the next step of the dialog, specify whether Cantata is allowed to download additional data from the Internet about the individual songs and albums and store them in the music folder. Data to be downloaded primarily includes lyrics and covers for albums. The configuration is now complete, and the wizard branches to the program window.

Settings

Cantata is somewhat reminiscent of the Rhythmbox audio player, which is often used on the Gnome desktop: On the left side of the window, you will find various buttons for managing audio files, playlists, and music sources. The center of the window is occupied by an empty list segment.

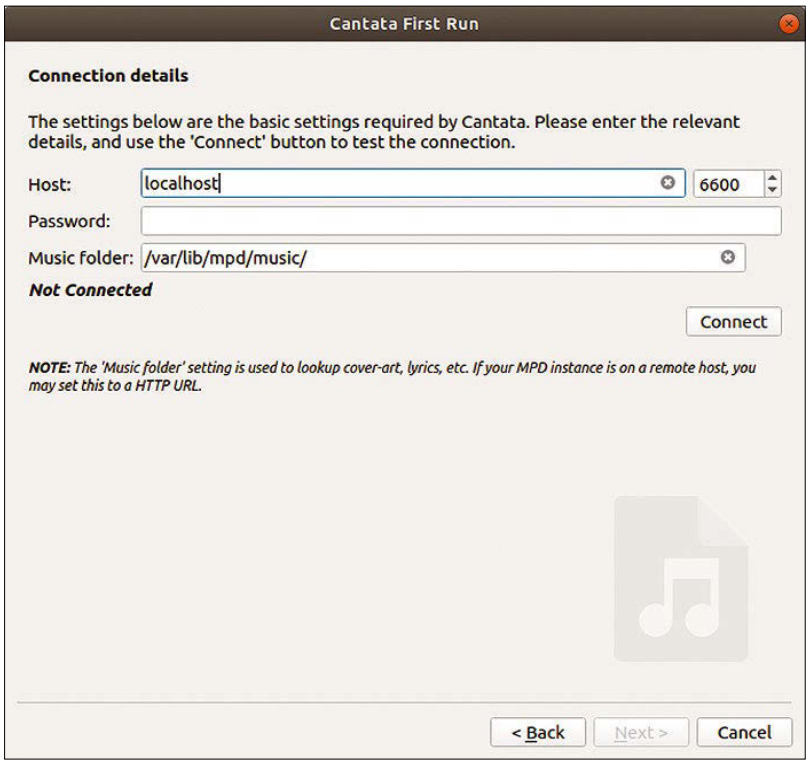


Figure 2: Cantata has a configuration wizard for easy setup.

On the right, in a large, at first still empty, area the individual tracks or sources appear in tabular form after selecting albums or lists.

Additionally, you will see a grayed out control bar at the top, which you can use later to play the individual tracks. On the far right next to the controls is a gear symbol, which opens another settings menu. Alternatively, you can access the configuration dialog by right-clicking on one of the buttons in the left sidebar of the software and then choosing the *Settings* option.

Pressing Ctrl+M displays a menubar at the top of the screen to facilitate individual operating steps. When you get there, you can also access the configuration menu from the *Settings | Configure Cantata* entries. This menu arranges the individual configuration groups in a sidebar on the left, and you can adjust the numerous options in a larger area on the right (Figure 3).

Static IP

On a home network, the router repeatedly assigns new IP addresses to the connected computers. Therefore, you will want either to talk to the MPD server in Cantata via the hostname or to configure the router interface so that the server always receives the same IP. Alternatively, enter a static IP address on the MPD server system itself. Be careful not to interfere with the router's DHCP server. This usually avoids a defined IP range, which you can then use for static assignments.

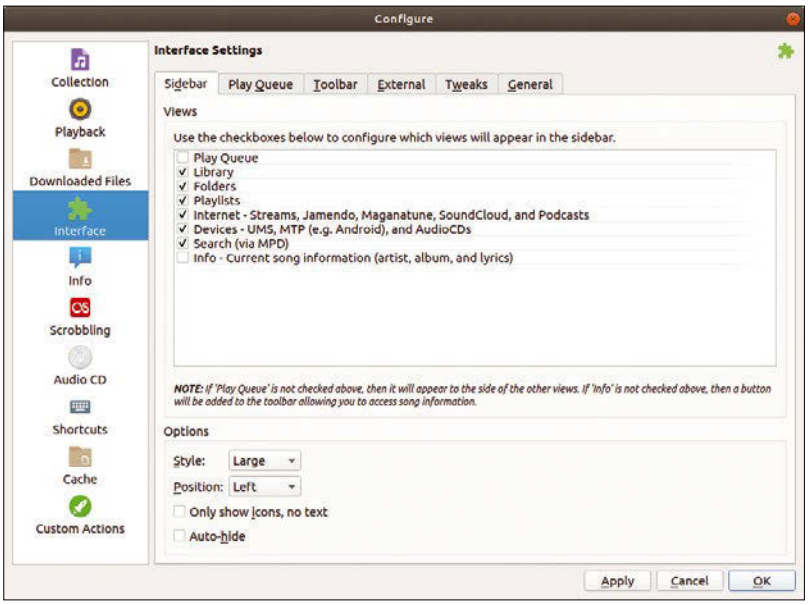


Figure 3: Granular adjustments of Cantata are made in a clear-cut dialog.

In the dialogs, you can configure settings for the individual groups in the sidebar of the main window as well as for the appearance of the application. For example, you can hide entire groups from the main window if you do not need them. Numerous sources for lyrics or cover images can be activated at this point. The modified options only take effect after pressing the *Apply* button at bottom right in the settings window. After entering all options, press *OK* to close the dialog.

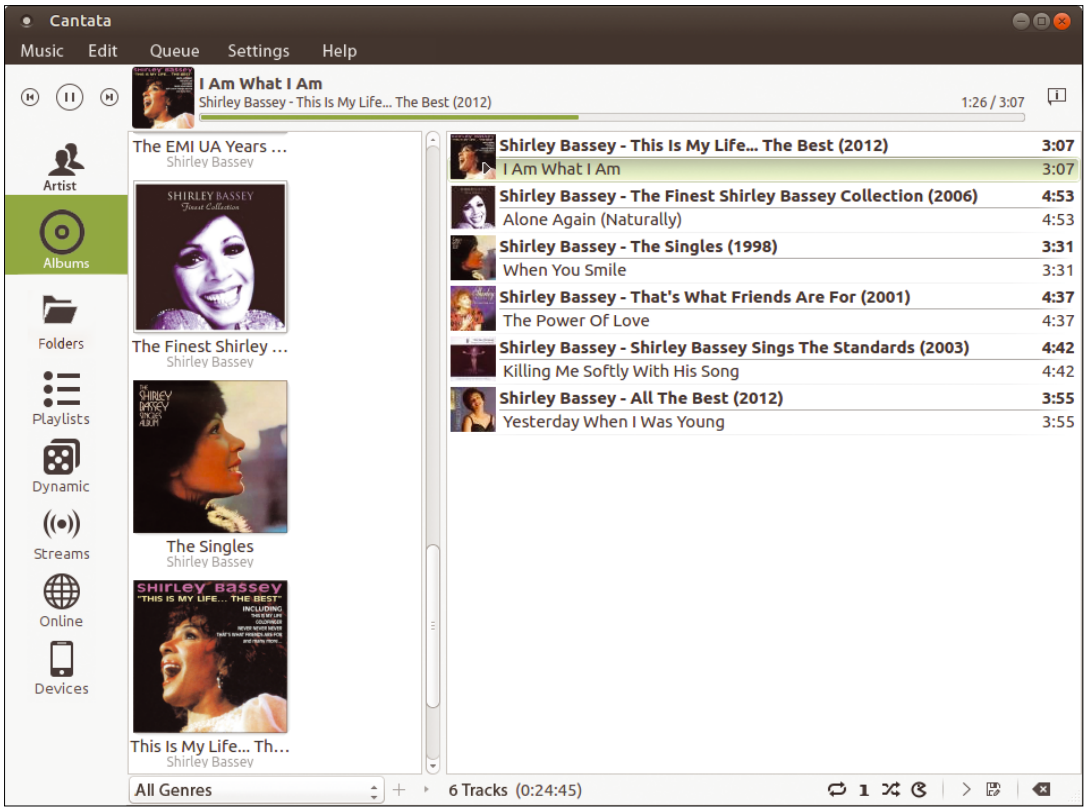
Disk Jockey

The main controls on the left side of the program window are used to create and manage playlists. To begin, click *Artist* to display the names of the tracks imported from the hard drive in the central window pane. A click on the second button instead shows the imported *Albums*, which Cantata displays with the cover images, if it finds them on the hard drive or on the web. Clicking on one of the albums opens a list of tracks in the central pane.

Clicking on the plus symbol to the right of each track transfers it to the temporary playlist. The matching title then appears in the right-hand window pane. Clicking on the *Play* icon in the upper left corner of the taskbar in the program window starts playing back the current playlist. The horizontal bar to the right shows what is currently playing. Playback of the track on the MPD server starts immediately without latency (Figure 4).

To generate a permanent list from this temporary playlist that can be recalled at any time, first select the individual tracks and then right-click to open a context menu. Click *Add to Playlist* and then *New Playlist* and enter the name of the new list in the dialog that then appears. Then use the *Playlists* button on the left side of the window to open the playlist. Now you can fill the list with tracks in the same way as you did the temporary playlist. In the context dialog of the music selection, use the *Add to Saved Playlist* menu item.

Figure 4: Cantata makes creating playlists easy.



Playlists can also be edited. To help you do this, you will find various options in the context menu of the titlebar, which you can access by right-clicking on a selected track. Use the *Edit Track Information* entry to edit the metadata of individual tracks and *Remove Duplicates* to remove duplicate tracks. The *Sort By* submenu lets you sort the list by artist or album, for example. You can remove unloved pieces of music with the *Remove* option, and clear the entire list with the *Clear* entry. Click *Save As* to save the current playlist under a new name.

Localization

Cantata not only lets you manage and play your private music collection on the MPD server but also integrates external sources. The *Streams* menu group can be used, for example, to integrate web radio stations into your audio collection. Additionally, Cantata lets you integrate online services such as SoundCloud [3] or podcasts into the software. Even external devices such as smartphones that transfer audio data to Cantata via the MTP protocol find their place in the software.

To access these sources conveniently, create a corresponding group in the sidebar if it is still missing by opening the preferences dialog. Select the desired option in the *Interface* group in the *Sidebar* tab and enable it by checking the box. Cantata copies this group to the sidebar of the main window. Then, select one of the active providers from the *Streams* group. Cantata already offers some streaming services, and you can add others from the settings menu in the *Streams* dialog. After loading the streams, you can include individual streams in the playlist in the same way

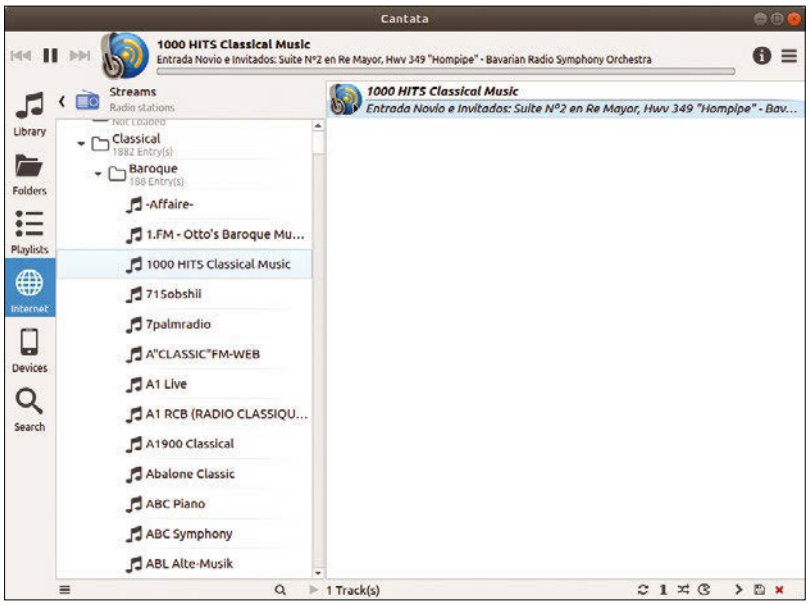


Figure 5: You can also play web radio broadcasts with Cantata and the MPD server. Note that the interface of this version of the software places Streams under the Internet icon.

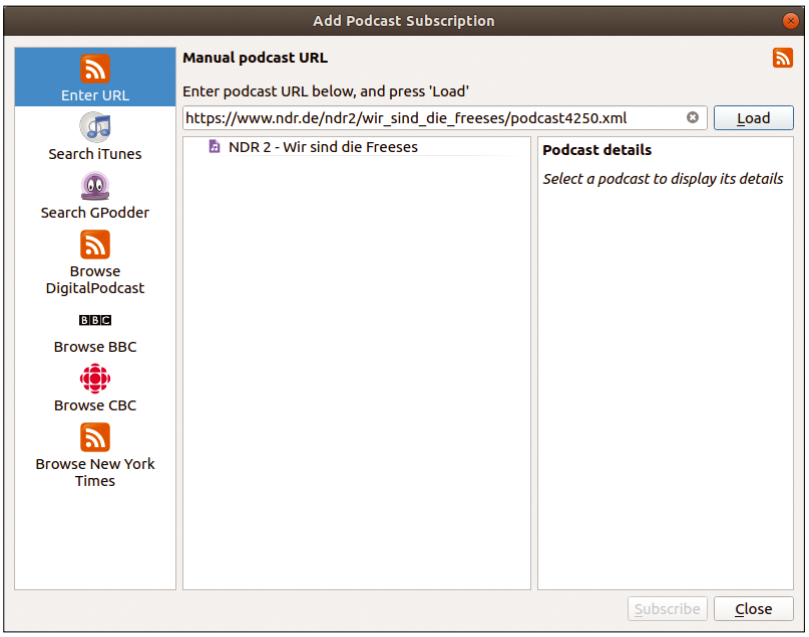


Figure 6: Playing podcasts is no problem in Cantata.

you manage a music collection. You start the desired stream just like a track; Cantata immediately loads it and plays it on the MPD server (Figure 5).

You can also integrate podcasts into the system under the *Internet* group; this opens a dialog for subscribing to podcasts. You will find many preset podcasts here, but you can also enable your own subscriptions in the *Enter URL* option. All you need is the corresponding URL, which you will find on the respective website. After entering the URL in Cantata, press the *Load* button on the right-hand side of the window to subscribe to the podcast. The podcast details then appear on the right side (Figure 6). In the main window, you can then open the desired podcast like any song.

Info Center

Cantata provides background information on artists, albums, and lyrics. To this end, the application accesses a number of online services. During track playback, you can hide the list window by clicking on the info bubble at top right and display an informational window instead.

In its three-part view, you will find biographical information for each artist on the left, a cover photo of and information about the current album in the middle, and the lyrics of the track you're currently playing on the right. On moving to the next title, the application automatically retrieves the new information and displays it immediately (Figure 7).

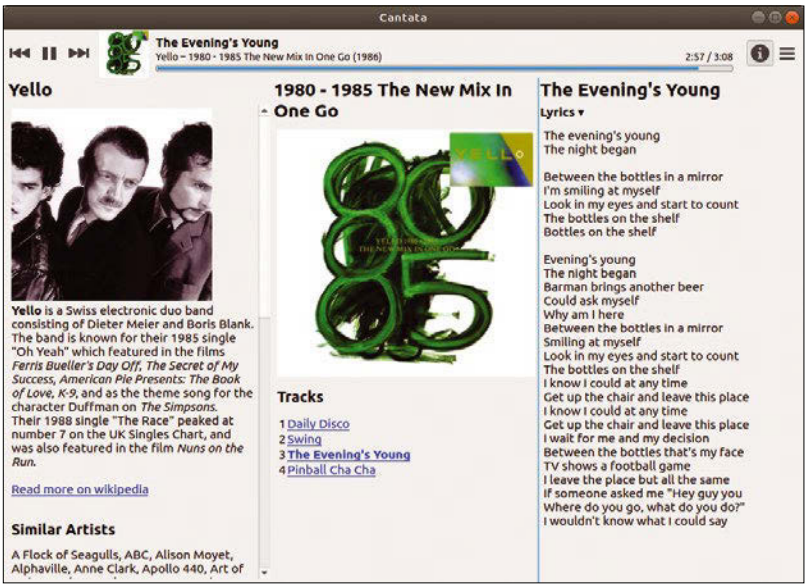


Figure 7: A full set of information about the artist, the album, and the track is available at the click of a mouse.

External Playback

Alternatively, you can configure the MPD server to send content to a remote streaming server such as Icecast 2 [4], which then distributes it. In this scenario, the MPD server acts as the source client. The program then does not play the content through the sound card of the MPD server and devices connected to it (e.g., hi-fi systems or sound systems), but through the sound card of any client that retrieves the contents of the streaming server.

Most modern audio players under Linux are capable of receiving streams off the web, for which you need a computer running an Icecast 2 server. On the MPD server, adjust the `/etc/mpd.conf` configuration file; to do so, enable the default configuration for the Icecast server in the `Audio Output` section by removing the hash signs from the start of the corresponding lines. After restarting the MPD server, it contacts the Icecast server and then streams content to it.

Another option is to access the MPD server from external clients in a web browser. The developers have integrated a small web server into the system, which you can also enable in the `/etc/mpd.conf` file. The web server is configured in the *Audio Output* area and is usually used with Ogg Vorbis or an MP3 encoder. The ***bitrate*** or ***quality*** option lets you adjust the quality of the playback to suit your needs. Please note that only one option can be active. Then, call the stream in the web browser on a client on the network by entering the matching URL *http://server_IP:8000*.

Conclusions

Thanks to the MPD server and Cantata duo, you can change the way you play back your favorite audio in an extremely flexible way. The MPD server is configured in a single file whose entries are well documented by the developers. Cantata is intuitive to use and does not require much training. ■■■

Info

[1] MPD server: *https://musicpd.org*

[2] Cantata: *https://github.com/CDrummond/cantata*

[3] SoundCloud: *https://soundcloud.com*

[4] Icecast 2: *http://icecast.org*

Linux Magazine is your guide to the world of Linux. Look inside for advanced technical information you won't find anywhere else!

Expand your Linux skills with:

- In-depth articles on trending topics, including Bitcoin, ransomware, cloud computing, and more!
- How-tos and tutorials on useful tools that will save you time and protect your data
- Troubleshooting and optimization tips
- Insightful news on crucial developments in the world of open source
- Cool projects for Raspberry Pi, Arduino, and other maker-board systems

If you want to go farther and do more with Linux, subscribe today and never miss another issue!

Subscribe now!

shop.linuxnewmedia.com/subs

GET IT NOW!
FAST DELIVERY WITH OUR PDF EDITION

À la Carte

Cutting recipes out of magazines and attaching them to the fridge is a thing of the past. Today, Gnome Recipes is your friendly kitchen helper. BY MARIO BLÄTTERMANN

Satirist Ephraim Kishon once confessed "food is my favorite food." Our earliest ancestors might have focused more on food than on enjoyment because of the considerable amount of time it took to procure it, but today's picture is different. In a fast food age, many people treat cooking, baking, and enjoyment of the results as pleasant leisure activities. Numerous specialized magazines bear witness to this, and many print publications feature a recipe page.

In the long run, this abundance of sources results in loose recipe clippings that threaten to fall out of your cookbook, often exceeding the book's thickness. If you make room for a laptop in the kitchen, the new Gnome Recipes program can save you not only from this avalanche of recipe clippings but a huge amount of time, from writing shopping lists to setting the timer to converting portions [1].

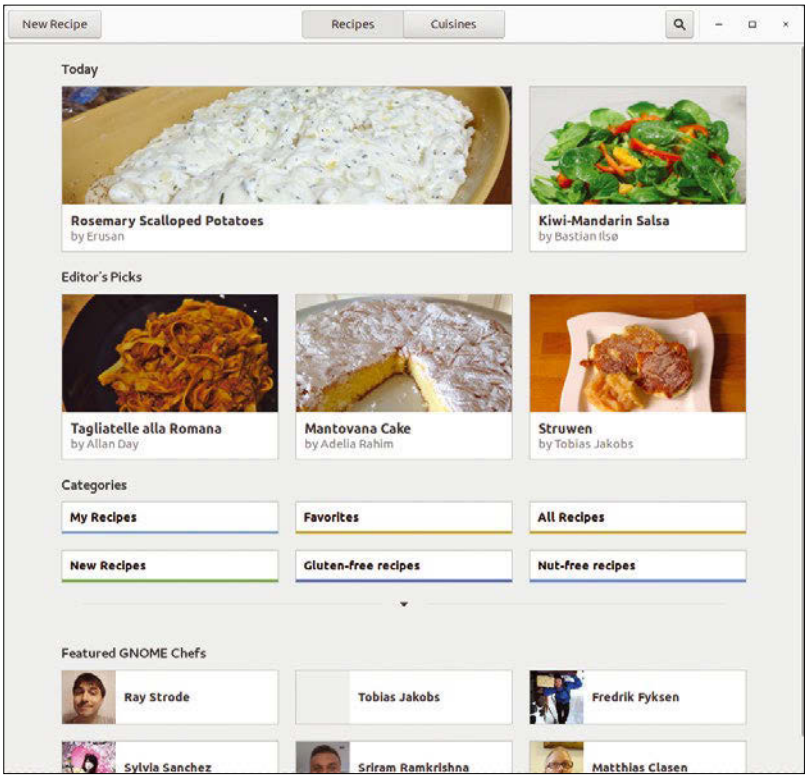


Figure 1: Gnome Recipes welcomes you with a clearly structured screen in the current Gtk3 design.

Delicious Pictures

The first time you open the program, your mouth might water: The main window (Figure 1) shows recipe pictures, lets you select recipes by category, and shows information about the contributing hobby chefs at the bottom. To start, simply click on one of the photos.

The application describes dishes in detail on the corresponding recipe pages (Figure 2), such as Struwen, a very tasty yeast dough pancake known in Germany's Münsterland and the Lower Rhine region. There is even a reference to a Wikipedia entry, but it cannot be clicked or copied.

If you mouse over the image, arrows appear that take you to more images of the dish or directly to its preparation. If the photos appear somewhat blurry at first, you do not need to see your local optician: The program first loads only a tiny preview image and then retrieves a higher resolution image off the web. For many recipes, the application displays small icons below the illustration that denote allergy information and dietary restrictions (for vegetarians or people who follow religious dietary regulations).

Click the *Print* button at the bottom right to create a hard copy of the entire recipe, including picture, list of ingredients, and instructions. The same applies for the shopping list: Click *Buy Ingredients* at bottom left to add the ingredients; then, click *View shopping list* to call up the shopping list. Most recipes are designed for four servings; if you are cooking for fewer people or for a larger group of Linux fans, the number of servings on which the shopping list is based can be changed in the *Yield* field.

However, the shopping list printout still needs some optimization (Figure 3): The quantity specifications and their conversion within the program leave some questions. For example: Why does the application calculate salt and sugar by volume, and why are raisins, yeast, and oil on the list as bulk? The developers still have to work on localization here.

Once you have everything you need, use the *Start Cooking* button to begin preparation. The

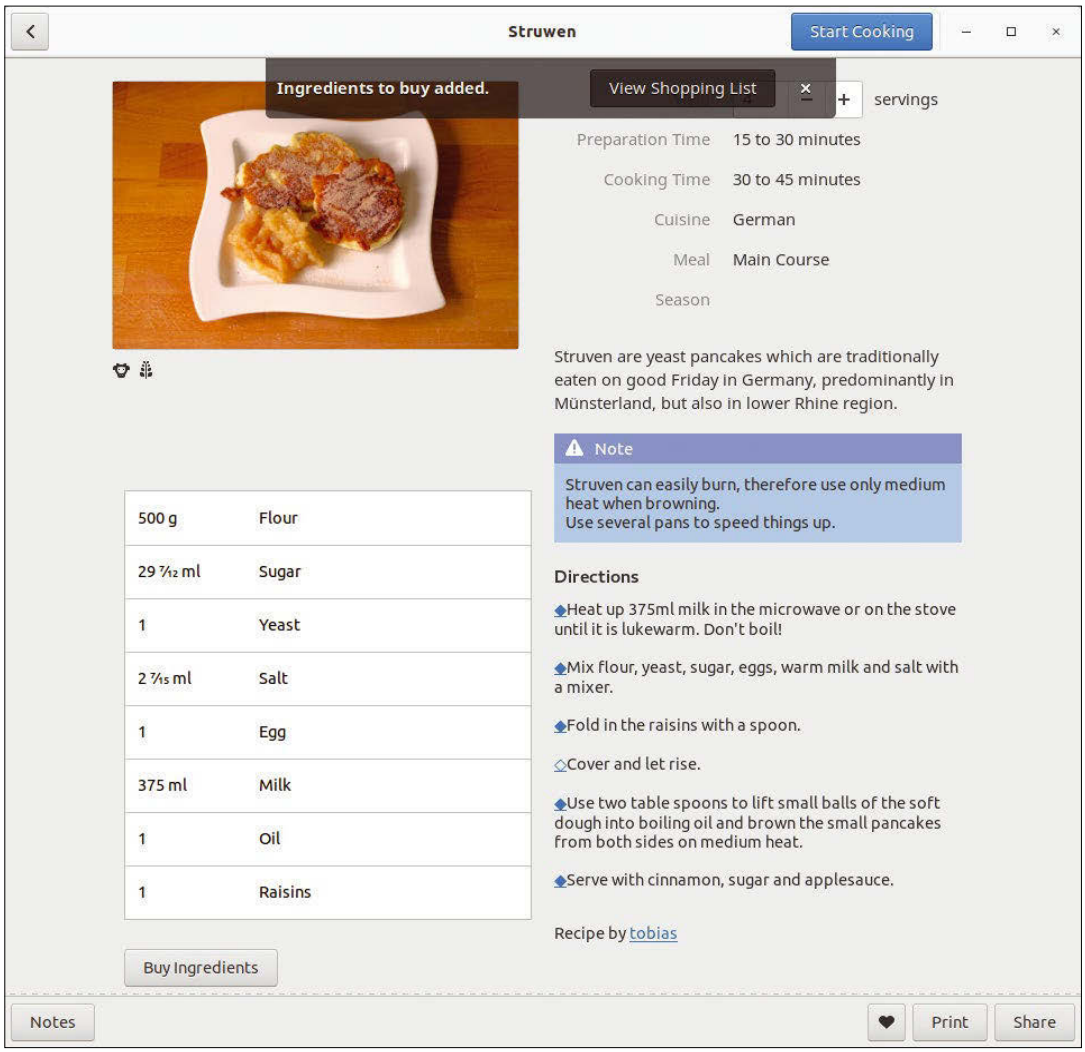


Figure 2: The recipes – like Struwen, a typical Good Friday dish – come from the community.

screen changes to full-screen mode, and the arrows on the right and left switch between the individual steps.

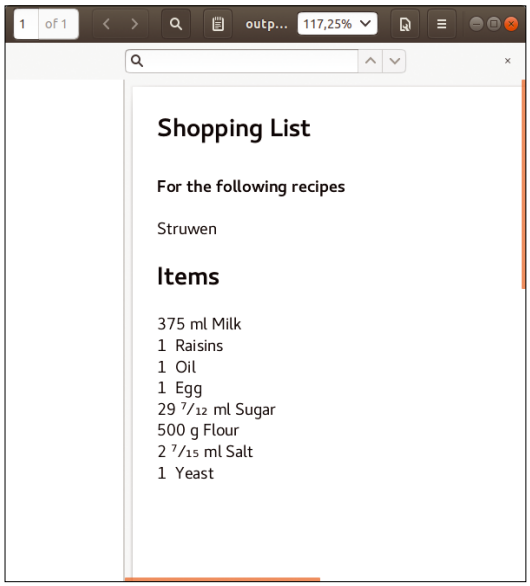


Figure 3: One raisin, please? The shopping list can cause confusion.

If you do not have a touchscreen device, it is better to use the keyboard while cooking. A single press on any key moves to the next step, a double-press to the previous one. Press Esc to return to the recipe.

A help window, which appears in the first window, outlines the keyboard controls (Figure 4). You can even use these controls to skip steps, such as the baking time (in this example, 30 minutes), but you will still be prompted to set the timer.

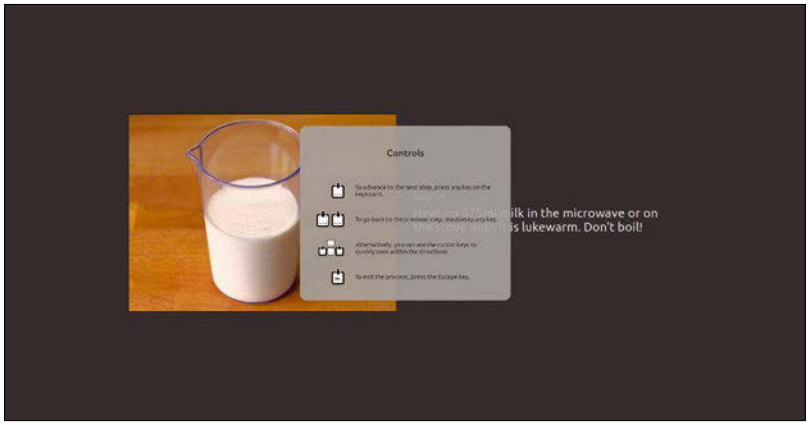


Figure 4: When you are cooking, recipes switch to full-screen mode, which can also be controlled on a touchscreen.

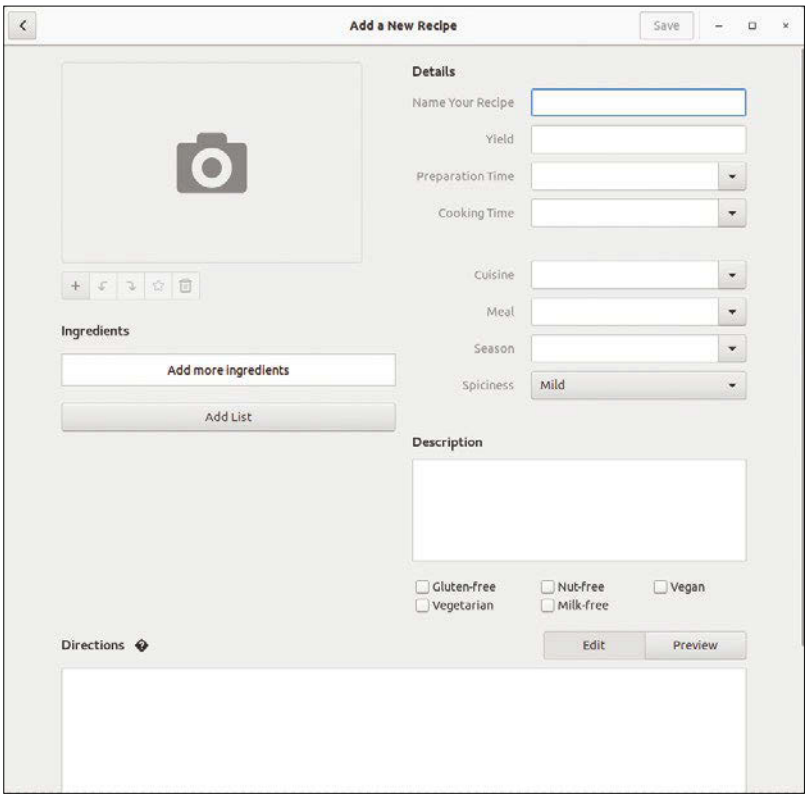


Figure 5: The community is happy about every new recipe: Just add your favorite dish.

Your Contributions, Please!

To share your recipes with the community, first add information about yourself. The *Chef Information* entry top left in the application menu (or in the upper bar of the Gnome Shell) takes you to the corresponding input fields. The Application applies the username set in Gnome; you can add some additional information, but you can also discard it and replace it with an alias.

Use the *New Recipe* button in the top left corner to open the input window (Figure 5). Most input fields are self-explanatory. You do not need to fill in information on, for example, *Spiciness* or *Season*. The checkboxes at bottom right are definitely helpful for food sensitivities, vegetarians, or vegans.

After clicking *Save*, your recipe is available in your own collection. Use the *Share* button on the right to make the recipe available to other users (Figure 6). The default setting is *Share with a friend*; alternatively, you can share with the community with the *Contribute to Recipes project* checkbox.

Both variants require a configured mail client. An email provider's web interface is not enough – unless, for example, you have set up Gmail with Gnome Gmail [2] as a dummy for a native client. If no suitable program is available, you still have the option to output the recipe as a PDF document and send it as an email attachment. However, this does not let you pass the recipe on as a native file that can be imported into Gnome Recipes on another computer.

Theoretically, exporting the recipe from the Application menu is still possible. This was successful, but the program crashed reproducibly. Here the developers have to make improvements, either by stabilizing the current *Export* routine or by offering the recipe directly as text that can be copied and pasted into email.

Conclusions

Gnome Recipes version 2.0.2 from December 2017, which was the version tested here, is a pretty smart apprentice for the kitchen. Crashes, nevertheless occurred, for example, when reactivating a previously closed shopping list or when exporting recipes. No other serious problems occurred.

The program follows the current Gnome user interface design somewhat, so it might cloud the picture in other work environments but should not prevent you from using the software (see the "Setting Up Gnome Recipes" box for installation instructions). The large main window is a bit annoying, though. Although Gnome applications should work well on netbook displays with 1024x600 pixel resolution, my test device's 900-pixel horizontal resolution screen was not enough to display the window without scrollbars.

Although recipes are often only available in English or with incomplete localization, hardly any other wishes remain unfulfilled with regard to the variety of templates. As always, free software thrives on participation. So make every effort to make your favorite creations or regional classics accessible to a wider audience. You can also contribute to the translation of the program and the recipes [3]. ■■■

Listing 1: Installing Gnome Recipes

```
$ rm -rf build
$ meson --prefix=$HOME/.local/build
$ ninja -C build
$ ninja -C build install
```

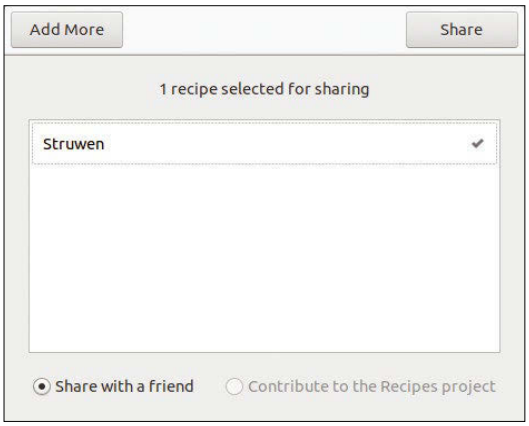


Figure 6: If necessary, you can email recipes to friends and acquaintances; otherwise, use the *Share* button.

Setting Up Gnome Recipes

Initially Gnome Recipes had a hard time making its way into the package sources of common distributions, but now a current version can be installed on Debian "Sid," Arch Linux, Mageia Cauldron, and openSUSE Tumbleweed. Ubuntu has had Gnome Recipes in its repositories since "Zesty Zapus."

To install from the source code, compile the tarball [4], which can be downloaded from the Gnome project as shown in Listing 1. If you only want to test the application, you do not have to execute the last command; the program also runs directly from the folder. However, this requires a trick; a single double-click on the binary file will not work. The command

```
$ GSETTINGS_SCHEMA_DIR=build/data ↵
  ./build/src/gnome-recipes
```

works if you call the program in the root directory of the unpacked tarball.

Gnome is currently gradually converting the build system from GNU Autotools to Meson, on which Gnome Recipes is based. This means that the three-step Autotools trick, which used to be common in Gnome, no longer works. To compile Gnome Recipes to work on your system, you need the `meson` and `ninja` commands. The latter can usually be found in the *ninja-build* package, on Mageia and Arch Linux in `ninja`. You also need to install the developer packages for Gtk3, Gnome online accounts, REST, JSON-GLib, and libsoup on your system.

Alternatively, use the Flatpak package [5], which the developers themselves offer on the project's homepage, including a nightly build of the current state of development. The package with the current version worked very well on Fedora 27 in our lab. However, Flatpak packages require a lot of disk space; including the necessary run-time environment, the installation took more than 200MB.

Info

[1] Gnome Recipes: *https://wiki.gnome.org/Apps/Recipes*

[2] Gnome Gmail: *https://davesteele.github.io/gnome-gmail*

[3] Translations: *https://l10n.gnome.org/module/recipes/*

[4] Source code: *http://ftp.gnome.org/pub/GNOME/sources/gnome-recipes/2.0/gnome-recipes-2.0.2.tar.xz*

[5] Flatpak package: *https://gitlab.gnome.org/GNOME/recipes/raw/master/flatpak/gnome-recipes.flatpakref*

FOSSPicks

Sparkling gems and new releases from the world of Free and Open Source Software

Graham tears himself away from updating Arch Linux to search for the best new free software. BY GRAHAM MORRISON

Image editor

Gimp 2.10

If there's one Linux and open source application that surely needs no introduction, this has to be it. Gimp is as infamous for aping the functionality of proprietary software as it is for its name, but many of us have been relying on it to edit our images for over two decades. Taking us from the pre-Google era of GeoCities, right through to the selfie-loving modern age, Gimp is one of the cornerstones of the Linux desktop, and despite recent excellent competition from the likes of Krita – an application designed mostly for illustrators – Gimp has remained the tool of choice for healing images, adding drop shadows, and generally editing images for anything more than a crop. But development has always been painfully slow, mainly because the project has a tiny team and little funding, which is why it's absolutely wonderful to include a new major version on these pages.

The result of more than six years of work, Gimp 2.10 is a major release both because it crams in so many features and because it simply exists. Fresh features include some excellent new tools (a warp transform, a unified transform, and a handle transform), the completion of its port to GEGL for acceleration and high-bit-depth processing, metadata editing, high-definition display support, and new themes. The warp transform lets you draw pixel manipulation onto your canvas and is ideal for small smudgy fixes or for warping a checkerboard. The unified transform is something Gimp has only hinted at with previous versions. It adds the ability to rotate, scale, shear, and change the perspective of an image using handles on the outside of the selected area. Just about every other image manipulation tool has had this since the dark ages, and it's essential for making small changes without resorting to a menu and entering values into a field. Finally, the handle transform is our favorite, because it also does a great job of showing off the new hardware acceleration. You add handles to your canvas in the same way you might add plungers to a window pane. The handles are then manipulated to move and transform the canvas. The best thing about the handle transform, and the others, is that it now all updates so quickly thanks to the new acceleration, which is something older versions of Gimp really suffered from. It means you can now load an image from a modern phone and manipulate it in real time with a normal PC, which you couldn't do without a huge time lag before.

1. New icons: Modern symbolic replacements for the old icons with less contrast. 2. HiDPI: Gimp now looks great on every desktop, whether it has a high or low pixel density. 3. Handle transform: Drop anchors onto your canvas and use these for multidimensional warping! 4. Warp transform: Smudge the canvas in mathematical ways. 5. GEGL acceleration: Many of the processing filters are now much faster and work in real time. 6. Canvas rotating, flipping, and symmetry painting: Deluxe Paint for the 21st century. 7. Extra formats: Load and save OpenEXR, RGBE, WebP, and HGT files. 8. Color management: This is now a core feature and includes most widgets.

The new themes also look good, and the emphasis on dark themes is a sensible choice, as you need the contrast to be most visible in the images. The high DPI support finally means you can use Gimp on screens that many photographers and artists use precisely because of their pixel density, and there's better Photoshop importing, with added support for new formats. But the best feature of all is that this is a big update to Gimp – we hope, the first of many to come.

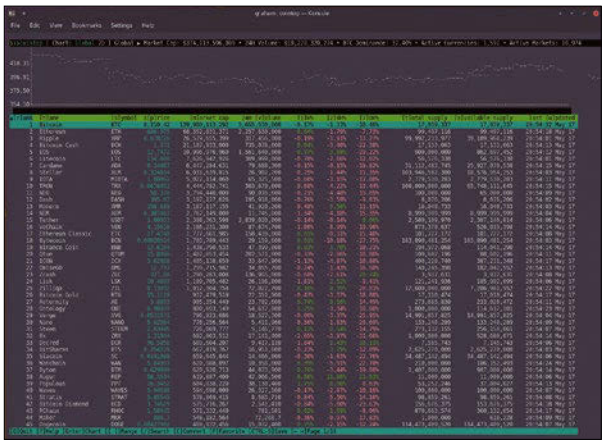
Project Website
https://www.gimp.org

Cryptocurrency tracker

cointop

It may have taken 10 years, but the price and trading explosion of late 2017 pulled the likes of Bitcoin and Ethereum into a whole new realm of attention and scrutiny. And yes, there's a lot of legal and moral ambiguity surrounding cryptocurrencies, but they're fundamentally a series of open source algorithms that implement a chain of trust. It's we humans that place value on them, whatever that happens to be. That makes them legitimately interesting if you're interested in technology and a legitimate waste of time if you want to track their value and their trends – which is where the wonderful `cointop` comes in.

When Hollywood gets around to dramatizing Bitcoin in the same way it has Jordan Belfort and Wall Street, it will show a wall of terminals and a hacker running `cointop`. It looks exactly as you'd expect a tracker for deviant currencies to look. Run from the command line, it lists the top cryptocurrencies sorted by market cap (by default) in the same way `top` lists your system processes sorted by CPU load. The top bar, like the ticker on a stock exchange, shows the 24-hour volume, Bitcoin dominance percentage, the total number of markets, and several other mind-boggling statistics. Beneath this you get a histogram that defaults to average price over time, changeable by selecting a currency with the cursor keys. Below this is a huge table giving you everything you need to know about each currency. It's much more effective than using a website and genuinely insightful because there's no opinion, no advertising, and no speculative figures – just the facts presented in a clear and data-rich way. If you have any interest in this kind of thing, even as a passive observer, `cointop` offers a brilliant peek into the burgeoning and entropic world of new currencies.



Better than a web portal, cointop keeps you unproductive at work by bringing crypto prices to the command line.

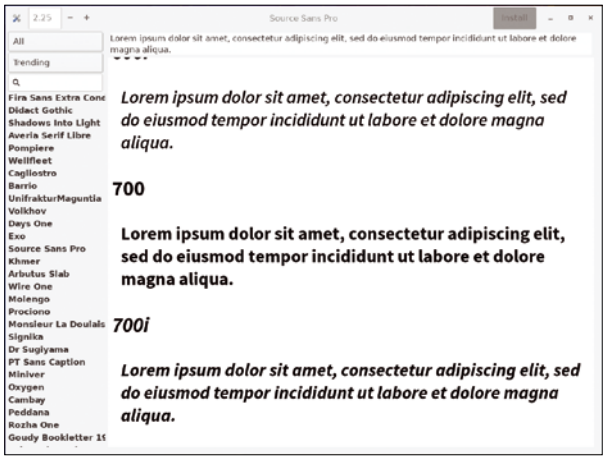
Project Website
https://github.com/miguelmota/cointop

Online font search

Font Finder

Fonts aren't something most users spend too much time thinking about. Most distributions include a good selection, with even some, such as Ubuntu, bundling their own fonts designed specifically for clarity and design aesthetics, but if you spend a lot of time with code or words, you'll know that the right font doesn't simply look good; it also helps ease eye strain and aid clarity. For those reasons, spending a little time trying out a few more options than those provided by your desktop or distribution is likely to reap some excellent usability, design, and health benefits.

KDE has a great font-browsing tool that will let you preview fonts, as well as let you install them either locally or globally with a single click – and you can even do this from Dolphin's right-click menu when more than one font is selected. But there isn't always an easy way of browsing fonts you may potentially want to install. Font Finder could be the solution; it acts as a kind of application store for the free fonts hosted on Google's font archive. It's also a lovely Gtk3 application, with a sleek minimal interface. There's a category browser, so you can easily search for monospace, for instance, and a search field that lets you filter fonts according to your criteria. Results are delivered in the main view, but critically, the results are previews of the fonts themselves, which is exactly what you need. You can change the size of the fonts and even render them onto a dark background. It's a brilliant way for looking through fonts you may wish to install. When you do want to install one, simply click on the *Install* button at the top right of the window.



Real-time search and font previews from Google's font archive, with a one-click install for those you like.

Project Website
https://github.com/mmstick/fontfinder

Task monitor

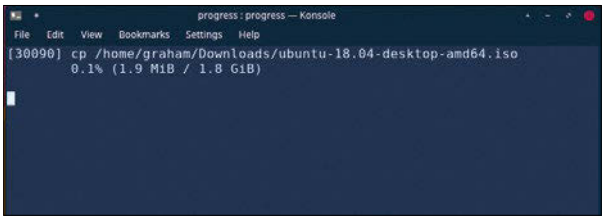
progress

Almost since the beginning of Unix time, Unix command-line tools have done very little to show their output. The bearded wisdom is that if you can't see any output from a command, then nothing has gone wrong, and you can assume whatever task you started has completed successfully. If you need error and success states, look for exit values and act accordingly. This principle actually works very well, and it would be lovely if it was adopted by modern technology, such as REST websites and convergent mobile apps. But it does leave you with some ambiguity over whether something is still running, and if it is, how long until you can go to bed and turn off your PC.

This is where `progress` can help. It works with many of the GNU core utilities that most of us take for granted. These core utilities include `cp`, `mv`, `dd`, `tar`, `gzip`/`gunzip`, `cat`, and many more. You simply run them alongside the `progress` command, either after the `watch` command to present a general overview of core utility commands or forked after the execution of one specific command. For example, you can monitor the progress of the humble `copy` command with

```
cp old.iso new.iso &
   newfile & progress -mp $!
```

with `progress` outputting the `copy` percentages as it completes. This is useful if you're copying large data files to a slow USB storage device on the command line, such as from a Raspberry Pi. Because it's simply monitoring the `/proc` filesystem for inputs and outputs on commands it knows, `progress` then probes `proc` to discover the status of those commands. It's a simple solution, but it works well and can be used in a huge variety of tasks, from monitoring a file or web server to the download status of something you've clicked in Firefox.



`progress` works with most core utility compliant commands on the Linux command line.

Project Website
https://github.com/Xfennec/progress

VPN

WireGuard

We should all be routinely using a VPN. They remove your reliance on a local ISP and shuffle the data gathering potential of any single point to one that's slightly less predictable. They also give you complete control over where you want to be located and how you want your IP address to be presented. In addition, they're ideal for encrypting a connection across unsecured networks, such as a cafe's WiFi. But many of us don't use a VPN, and there are various good reasons for this. The first is that you never know if you can trust your VPN provider – they could be worse than your ISP. Even when they promise not to log your incoming and outgoing connections, the state of the security industry means that they may simply not be permitted to say whether they are or not. The second is that when you do want to run your own VPN, it's difficult to find both a simple and a secure solution.

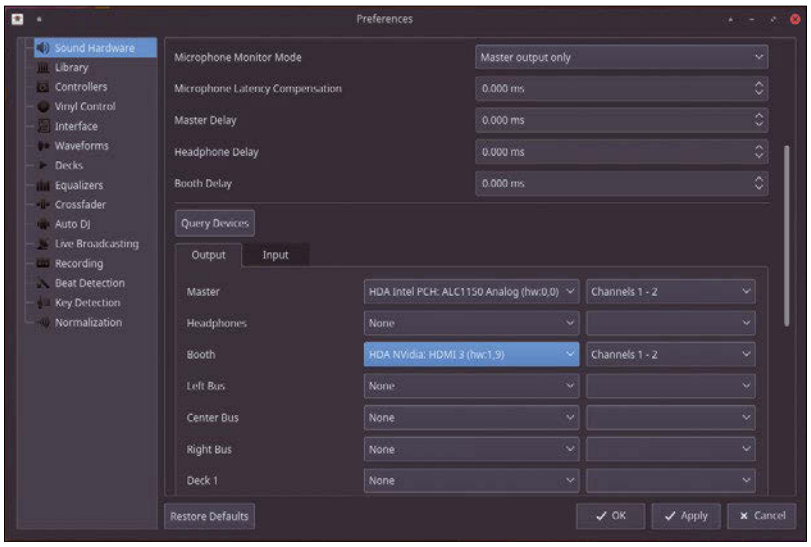
An OpenSSH tunnel and a SOCKS proxy can do a good approximation of a VPN, but WireGuard is a far better solution. The best thing about WireGuard, and why it potentially beats OpenVPN, is that it's simple in both configuration and implementation. While its 4,500 lines of code have yet to be security audited, it's a manageable number to check yourself. Like SSH, WireGuard's security is based on public key encryption. Installation is through the generation of private keys held in restricted configuration files and the creation of a WireGuard interface. With the connection up and running, VPN access is fast and transparent, and if you've ever tried to install and maintain OpenVPN, it's much easier to run.

WireGuard has yet to have a security audit, but the project is small enough that it's easy to understand.

Project Website
https://www.wireguard.com/

DJ software

Mixxx 2.1



Combine real hardware and controllers with great sound input and output configuration.

While Linux and open source music software can't compete with some aspects of the proprietary realm, in many ways it's more interesting because there's a real experimental bent to the way many open source tools are developed – maybe because developers don't need to worry about paying customers. The excellent people behind monome, for example, have announced a wonderful (and expensive) silver box for generating, processing, and controlling music and audio. But because it's built around a Raspberry Pi and Lua scripts running on SuperCollider, we can all get involved without the investment. Similarly, the developer behind the Organelle hardware platform has released Orac, an open source modular instrument hosting environment that can be used on a Raspberry Pi to create complex audio chains, all without a screen or proper computer in sight.

So, while we don't have all of the commercial hardware effects-emulating plugins or the industry-standard digital audio software, we do have some exceptional music projects for when we want to flex our creative muscles. One of these is Mixxx, a brilliant tool for any budding or even performing open source DJ, podcaster, or local radio producer. Version 2.1 has been two years in the making. It essentially replaces the turntables with two virtual "decks" into which you load your music from a library. In fact, this being the realm of virtual turntables, it only takes a few clicks to turn those two decks into four, all without selling your car or taking out a loan. But it's not quite all virtual either. You can use special timecode records on a real turntable that allow you to control the digital playback, complete with scratching sounds, just like a real '80s DJ.

Dance the night away using nothing but open source and your own music collection.

Rather than buying and storing your precious collection, however, you simply drag and drop music into the turntable area of the display. Multicolored waveforms are rendered into both the overview of what's playing and the audio loaded into each cue deck. You can even use beat detection, shown as vertical lines in the waveforms, to sync a cued track against the track that's playing. It's a relatively crude effect when there's a large tempo mismatch, but it works perfectly when two tracks have similar and clearly defined rhythms.

Effects are a big part of the playback chain. The Moog filter (unlikely to be officially endorsed) sounds sublime and is perfect for sweeping through one track before dropping into a crunchy beat on the second. There's EQ, a Bitcrusher, delay, and panning effects, many of which are new for v2.1 and all now syncing with the beat, obviating the need for external equipment or proprietary effects. For an open source application, it sounds fantastic and could easily be used professionally. For those DJs that like to interact or shout at their audience, you can see the live microphone input and even use servers like Icecast to broadcast on the Internet. This new version now scales on high-resolution displays, lets you resize the waveforms for extra clarity, adds new sampler configurations for effects and extra loops, and feels amazing. This is one of those rare breeds of open source applications: one that's as good or better than the proprietary competition, as well as creative, inclusive, and capable of genuinely professional results.

Project Website
https://mixxx.org/

Pass CLI GUI

rofi-pass

Use `pass` to manage your passwords and rofi-pass to access them easily from the command line.

We're huge fans of the `pass` password manager. The brilliant thing about `pass`, and what sets it apart from the likes of LastPass, is that you control its implementation. You use your GPG keys to encrypt minimal text files containing your passwords and usernames. Because you usually name these files after the web services they access, you can quickly navigate to the correct password when you need it. `pass` does this for you on the command line, and a GUI-like `qt-pass` does the same thing from the desktop. We're big fans of `qt-pass` too, but it needs to be mentioned that a recent release contained a catastrophic vulnerability that meant randomly generated passwords weren't being randomly generated at all, and instead were pseudo-randomly generated from a small and limited pool of possibilities. If you've used `qt-pass` to generate passwords, check which version you used and update your passwords accordingly.

The only problem with `pass` is that it has none of the convenience of the tools that automatically enter your password for you. This can be solved by scripting your own solutions, as users have done for the brilliant qutebrowser, but it's a job made considerably easier with rofi-pass, a `rofi`-based command-line user interface to your `pass` password repository. By simply running the command, you can browse your complete repository, select the service you want to access, and then enter your passphrase via your system's standard password entry form. You can even create new entries, all without needing to remember the specific pass syntax. It can be used on its own or within your own scripts, and it's a great solution for proto-browsers like qutebrowser that may not yet implement their own APIs or connect to popular password interfaces.

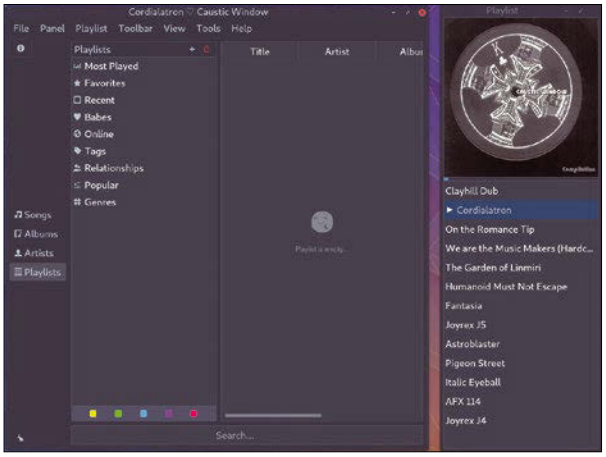
Project Website

https://github.com/carnager/rofi-pass

Music player

VVave

This month's music player was formerly called "Babe." Based on Qt, it's a rather neat little player that doesn't demand much of your system and manages to look great and support nearly every feature you'd expect. All of this might even start to sound familiar, because we've looked at Babe before – way back in May 2017 (*Linux Magazine*, issue 198). But Babe has been rapidly maturing, even maturing out of its infantile name, though sadly into a new one that's worse than a teenager's nickname – VVave (that's two Vs, for those with bad eyesight). This comes with a rebranding and a lovely website with a new design aesthetic and hopes of becoming a standard KDE application. The KDE integration is well justified now that the application has lots of KDE native notifications, use of KDE controls, and even the ability to send your music to your phone via KDE Connect – something that you need to sell your soul to do with iTunes.



Babe has become VVave, and although the name is no real improvement, the application is.

Even better is a new Chrome extension that goes alongside the main application. With this installed into your browser, you're able to add YouTube music videos to your collection with a simple click, adding the music to VVave, which enables you to play without the added distraction of moving pictures. VVave's main feature is still a brilliant one – you can remove all the distractions of the excellent playlist manager, or the contextual music details and links, and simply have a tiny window for playback, augmented perhaps by an album cover or maybe only the list of songs in the playlist. You've got lots of options on how many panels are visible and what the application does. It's really come a long way.

Project Website

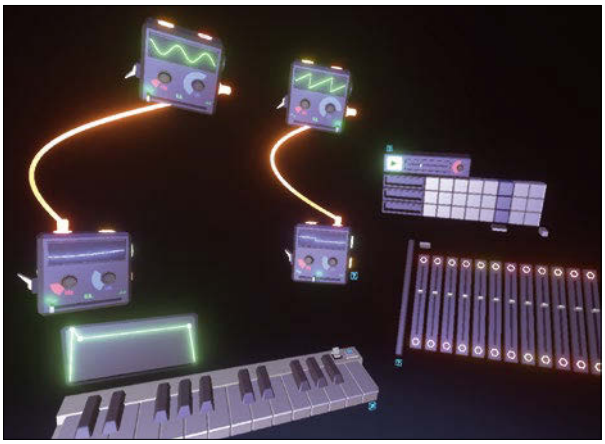
https://vvave.kde.org/

Virtual Studio

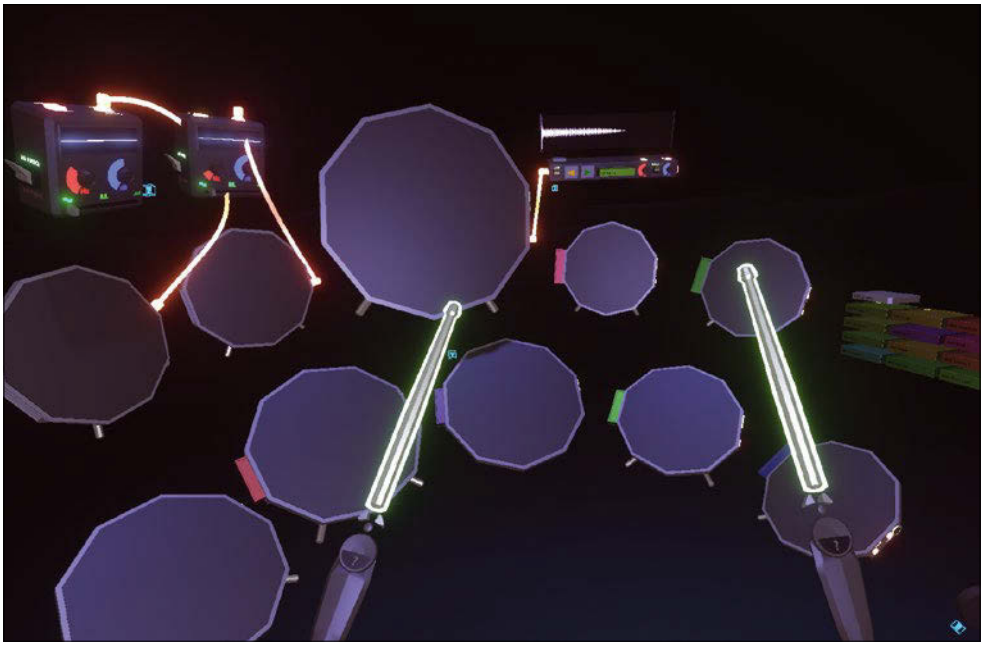
SoundStage VR

You can't have failed to notice that virtual reality (VR) has had something of a long-awaited renaissance. It hasn't been as dramatic or as life changing as many had hoped, but it has resulted in real and relatively affordable hardware, as well as a new Steven Spielberg movie. Unfortunately, despite the initial promise, the enterprise that literally kickstarted this renaissance, Oculus (now owned by Facebook), has failed to deliver any Linux support for its hardware, but its competitor, the Valve-backed SteamVR platform, has. If you have the money, you can buy an HTC Vive headset with a high-end Nvidia graphics card, install Steam and SteamVR, and play a handful of games in VR on your Linux box. It doesn't work as well as the Windows version, mostly down to input latency, but it does work. Strap the headset on and wave the controllers in the air, and you can play amazing games like The Talos Principle, or Windlands, or the incredible physics simulator, Universe Sandbox. These games are totally immersive, with the headset imperceptibly updating as you turn, walk, and jump to generate a three-dimensional view of your surroundings. It's still relatively crude and prone to inducing nausea, but there seems little doubt that when the technology improves and prices tumble, VR may become the standard computer interface.

Unfortunately, there isn't much room for open source in all this software. OSVR is a very promising platform that is open source and includes its own hardware, but there aren't many fully fledged games or applications that use it. One exception is Safespaces, a virtual desktop with a terrible name. The SteamVR platform is open, as in anyone can start writing for it, but it's not open source, and that means compromises will need to be made. If you are going to make a compromise, there's no better place to start with than SoundStage VR, a commercial application that was so brilliantly designed and effective in what it attempted to do that the developer was quickly hired by Google and the project abandoned. Before it was abandoned, though, the developer did the right thing and fully open sourced the project, which can now be downloaded and built for free (alongside the free, not open source, version of Unity).



Connect synth elements, effects, keyboards, mixers, samplers, drums, and sequencers together in any way you choose.



Open source virtual reality app, SoundStage VR, lets you build your own studio and make as much noise as you want.

There's even a pre-built binary if you don't want to build the code yourself. Just make sure you've got Steam VR running and run the executable from the command line. SoundStage will appear right in front of your eyes. It's not exactly a game, but this is an environment you can play in for hours. It's a virtual music and recording studio with a completely performance-driven modular approach. From a 3D palette, you can construct a complete virtual drum kit, for instance, linking drums in 3D space to prerecorded samples. Synthesizer elements can be connected with virtual wires, just as you might with a Moog modular synthesizer, all of which can be recombined, reconnected, recorded, and played in real time. You can even use external MIDI equipment and the JACK audio connection layer to get sound into and out of your virtual environment. It's a lot of brilliant fun, and if you're lucky enough to have a compatible VR headset, it's a must.

Project Website

https://github.com/googlearchive/soundstagevr

Needle in a Haystack

What grep cannot accomplish with LibreOffice and OpenOffice documents, a small odfgrep script can.

BY MARCO FIORETTI

If you have a lot of text files, slide shows, and spreadsheets on your computer, you will need, sooner or later, to know quickly which files contain certain words or sentences. You might also want to use that information to perform some other actions automatically, like sending email notifications or adding new records to a database. Sometimes, you can do this with the Recoll desktop search engine described in the previous issue of *Linux Pro Magazine* [1]. Should you, however, want something lighter or more flexible than Recoll, try `odfgrep`: It not only might work better, but also teach you other, very efficient ways to manage all your office documents.

What and Why

A really basic knowledge of the command line and Bash syntax is helpful, but not mandatory: The code is short and explained as accurately as possible, to help you learn some basics of shell programming, if needed.

In fact, the hardest part of this whole tutorial may not be the code itself, but figuring out why you might want to learn and use it. In a nutshell, learning how to search or otherwise process ODF files from the command line, with `odfgrep` or similar tools, can help you to become a much more productive desktop user, able to delegate to your computer many more otherwise very time-consuming tasks. That's it, really.

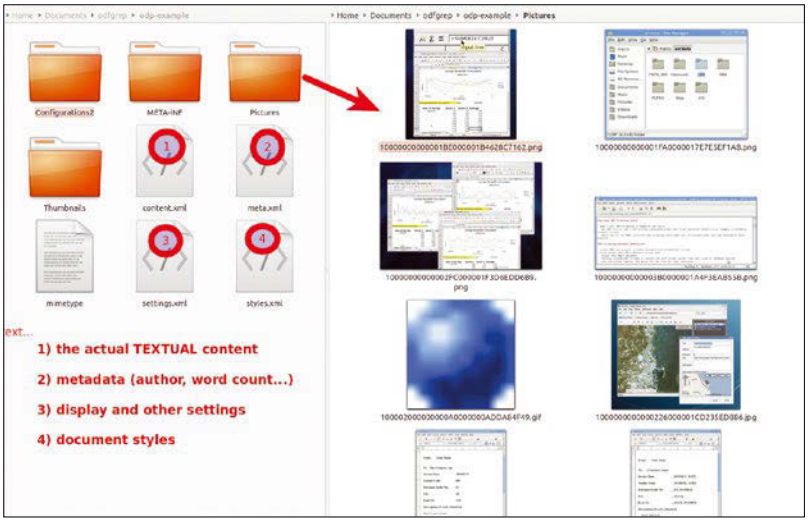


Figure 1: An ODF file is just an ordinary ZIP archive of images, folders, and XML files, each with its own, fully documented purpose.

What Is grep?

The Unix world, to which Linux belongs, has been using and improving tools for automatic processing of plain text files for decades. The `grep` command-line program is one of those tools and is one of the reasons why Linux is so great at text processing. By default the `grep` utility searches for lines that match a given pattern in all the files passed to it and then prints the lines or counts the occurrences. The `grep` options you are most likely to use are:

- `-c` (count): Print the number of lines matching the pattern.
- `-l` (list): Print only the name of each input file that contains the pattern.
- `-v` (invert match): Print only the lines that do not match the pattern.

All Hail ODF!

ODF is more than just a really open standard, which of course is an extremely important thing in and of itself. Compared with Microsoft Office file formats, or to almost any other format with comparable features, ODF is also very, very simple to analyze or generate automatically. In fact, as you can see in Figure 1, any ODF text, presentation, or spreadsheet is nothing but a ZIP archive of eXtensible Markup Language (XML) files, each with a predefined name and purpose, and pictures. XML is very verbose, but it is plain text, with tons of Free Software libraries, programs, and documentation to easily process it. At the end of this tutorial, for example, I include a link that contains my own little scripts for automatically generating ODF invoices or slide shows.

How odfgrep Works

If you thought that an `odfgrep` script would have to be very complicated, think again. The code in Listing 1 is the `odfgrep` that I use from time to time on my own GNU/Linux computers, and it is less than 30 lines.

There are surely many other ways to write an `odfgrep` script, and many of those ways may be faster than the example here or handle some weird combinations of search patterns and ODF files better. Personally, however, I have not had any problems yet, and the simple `odfgrep` discussed here should be enough for the needs of the great majority of Linux desktop users. Its high-level flow diagram (Figure 2) is easy to describe:

1. Get the list of all the files to analyze from the user.
2. Figure out which of those files are in ODF format.
3. Make a temporary plain text version of each of those files, each in the same folder as the original.
4. Pass to the standard `grep` the same options specified by the user, but a different list of files, in which plain text versions of the ODF files are used.
5. "Massage" the output of `grep` on the fly so that the user sees the names of the original ODF files.
6. Remove all the temporary plain text files created in step 3.

I look at the assumptions behind this algorithm, its limits, and some ways to expand it at the end of the tutorial. For now, I'll just look at how the code that implements it works, line by line.

Commands typed at a prompt are interpreted and executed on the spot, line by line, by special programs called "shells" in the Linux world. You can save long sequences of commands in files that a shell may then execute automatically, one line at a time. These special files are called scripts, and `odfgrep` is just that: a script.

The weird stuff on line 1 is the standard header of every shell script. The two initial characters ("shebang" in Unix slang) mean that the file uses the syntax of the default shell on GNU/Linux systems, called Bourne Again Shell (Bash), and therefore must be interpreted by the `bash` program that is in the `/bin` folder.

Each shell script can receive options, or switches, that modify its default behavior. In Bash, those options are saved in the special variable called `$@`. Lines 3 and 4 copy all those switches, for readability, in two string variables called `OPTIONS` and `ODFOPTIONS`. The first will only be used to figure out which of the files that `grep` is to scan are in ODF format (line 6).

The `$ODFOPTIONS` variable is filled in line 4 with the customized file list mentioned above. In that line, in fact, `sed` (Stream EDitor) receives the original options and appends on the fly the string `.odfgrep.txt` to each occurrence of the `.odt`, `.odp`, and `.ods` file extensions.

In other words, if you asked `odfgrep` to find all the occurrences of "Linux" in two files called `thesis.odt` and `thesis-slides.odp`

```
#> odfgrep Linux thesis.odt thesis-slides.odp
```

Listing 1: odfgrep Script

```
1  #! /bin/bash
2
3  OPTIONS=$@
4  ODFOPTIONS=`echo $@ | sed -e
   's/\.\(odt\|odp\|ods\)\b/\.\1\.odfgrep\.txt/g'`
5
6  for FF in $OPTIONS
7  do
8      if [ -f "$FF" ]
9      then
10         case "$FF" in
11             *.odt|*odp*|*ods)
12                 odt2txt --width=-1 $FF > $FF.odfgrep.txt
13                 FILES2REMOVE="$FILES2REMOVE $FF.odfgrep.txt"
14                 ;;
15             *) # non-ODF file found, nothing to do
16                 ;;
17         esac
18     fi
19 done
20
21 grep $ODFOPTIONS | sed -e 's/\.odfgrep\.txt//'
22
23 if [[ -n "${FILES2REMOVE// }" ]]
24 then
25     rm $FILES2REMOVE
26 fi
27 exit
```

then `$ODFOPTIONS` would assume the value `Linux thesis.odt.odfgrep.txt thesis-slides.odp.odfgrep.txt`.

`sed` achieves this by substituting each occurrence of the text pattern between the first two forward slashes with the other pattern between second and third forward slashes. A complete `sed` tutorial would not fit (and be off topic) here, but you need to understand two pieces of line 4: The `\1` means "put here

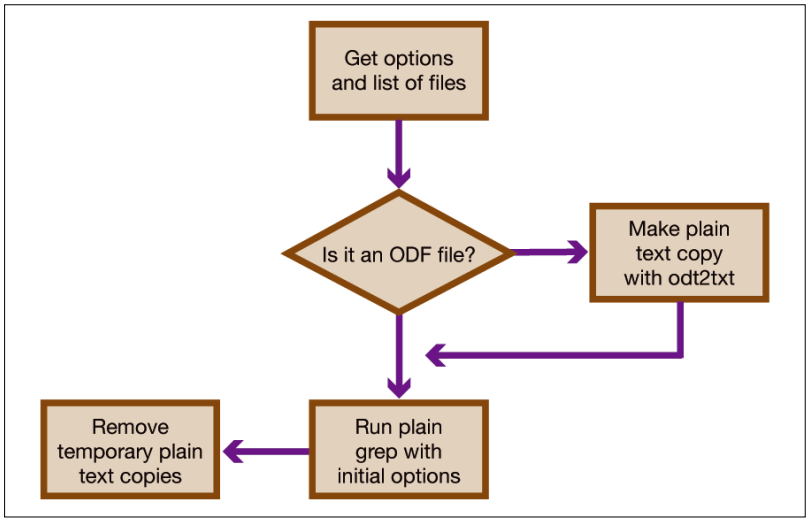


Figure 2: Main steps of the algorithm used to find text in ODF office documents, as if they were plain text files.

the string just found with the pattern in the set of parentheses to the left" (i.e., `.odt`, `.ods`, or `.odp`).

The `\b` pattern modifier makes `sed` only act on word boundaries: Without it, line 4 would modify a file name like `notes.odtconference.odt` to `notes.odt.odfgrep.txtconference.odt.odfgrep.txt`. Not a 100% bulletproof solution, since it would also work, say, on strings like `my.odt.notes.txt`. In practice, that has never been a problem for me.

The loop in lines 6 to 19 creates the plain text copies of each ODF file, saving their names (which are the same previously written in `$ODFOPTIONS`, remember?) in the `$FILES2REMOVE` variable. To do this, `odfgrep` copies the substrings inside `$OPTIONS`, one at a time, in the variable `$FF` (line 6) and looks at them. But nothing happens unless:

1. `$FF` is the name of an actual file (line 8), and
2. Its extension is `.odt`, `.odp`, or `.ods` (line 11).

In that case, and only in that case, the `odt2txt` utility (line 12) writes a plain text copy of `$FF` in a temporary file with the same suffix used in line 4 to build `$ODFOPTIONS` – that is, `.odfgrep.txt`.

Please note that, even if I just called it a "file name," `$FF` includes the path to a file (i.e., it may have values like `work/essays/phd-thesis.odt`). In this case, `odt2txt` would save the plain text copy as `work/essays/phd-thesis.odt.odfgrep.txt`, so it is in the same folder as the original. The same string is also appended, in line 13, to the variable `$FILES2REMOVE`, which is necessary for reasons that will be clear in a moment.

The `--width` option in line 12 tells `odt2txt` the width at which text lines should be wrapped. Its default value is `65` characters. Setting it to `-1` means "do not wrap lines." This adjustment is necessary because `grep` works line by line. If you were searching for a sentence like *Linux is great*, but `odt2txt` split it across two consecutive lines, `grep` would not find it.

Once the loop that started in line 6 ends, `$ODFOPTIONS` contains three types of "objects":

- The options and search patterns that the ordinary `grep` should use.
- The paths to all the non-ODF files passed by the users.
- The paths to all the plain text copies of ODF files generated in line 12.

The objects of the first two types are not modified in any way, because they were not file names with ODF extensions; therefore, the loop did nothing to them!

At this point, you can finally run `grep` with the `$ODFOPTIONS` (line 21), but with one trick: Filter its output with `sed` in a way that makes all the `.odfgrep.txt` strings disappear. This will make `odfgrep` always return the names of the original ODF files, instead of their plain text copies, which are the only ones that `grep` sees. Without that `sed` command, the output of `grep` could be something like

```
phd-thesis.odt.odfgrep.txt:
  Linux is great and I love it..
```

and this would confuse the users. The `sed` part of line 21, instead, transforms the output line above in this way, pointing to the original ODF file:

```
phd-thesis.odt: Linux is great and I love it..
```

After this, the only thing left to do is clean up (lines 23 to 26). Line 23 means "check if, after removing all whitespaces from the `FILES2REMOVE` variable, it has a number of characters greater than 0." If that is true, it means that at least one plain text file was created, and its name was appended to `$FILES2REMOVE` (lines 12 and 13). In that case, execute line 25, which removes all the files listed therein. Done!

As an example, Listing 2 shows the output of `odfgrep` on a test directory that contains several files of different kinds in different subfolders. The command says "show me all the lines containing the word *linux* (case insensitive) in all the files inside `testdir` and all its subfolders" (some lines were truncated for better formatting).

As you can see, `odfgrep` works and generates output in the same way as the standard `grep`, always returning the right file names, both on actual plain text files like `go-linux.txt` and on ODF slide shows (`mfioretti.odp`) or document files (`open-business-models.odt`).

In another example, I ask `odfgrep` to tell me how many times the word *politics* appears in the ODF text documents inside a certain folder:

```
#> odfgrep -c politics testing/references/*odt
testing/references/conference-proceedings.odt:2
testing/references/openness-essay.odt:3
```

Here, `odfgrep` found two matching documents in that folder; the word *politics* appeared two times in one file and three times in another file.

Listing 2: odfgrep for "linux"

```
#> find testdir -type f -exec odfgrep -i linux {} /dev/null \;

testdir/references/mfioretti.odp:Writer for several Linux magazines
testdir/references/mfioretti.odp:any Gnu/Linux distribution is OK
testdir/references/open-business-models.odt:Yochai Benkler, Linux and
  the Nature
testdir/notes/go-linux.md:what trouble? Why not check your data table
inside a spreadsheet or database? Because it's often...
testdir/notes/go-linux.txt:Linux(1) is the best kernel around
testdir/notes/go-linux.txt:Linux,1
testdir/notes/go-linux.txt:he actually said "Linux is the best kernel
  around"
```

Installing odt2txt and odfgrep

The `odt2txt` program [2] is present in the repositories of the main GNU/Linux distributions. On Ubuntu, for example, you can install it by simply typing:

```
#> sudo apt-get install odt2txt
```

To install `odfgrep`, first save the code in Listing 1 (except the line numbers [3]) in a plain text file called `odfgrep` with the use of an editor like Gedit, Kate, or the venerable Vi or Emacs. Then, copy that file to a directory (e.g., `/usr/local/bin`), where all users of your computer can access it, and make it executable with the

```
#> sudo mv odfgrep /usr/local/bin
#> sudo chmod 755 /usr/local/bin/odfgrep
```

commands.

Caveats and Limits

The `odfgrep` script explained here is simple but very useful, provided you acknowledge some of its limitations or underlying assumptions.

The first things to know are about folder and file names. Depending on the language settings of your computer, this `odfgrep` may fail on names containing characters that are not ASCII alphanumerical characters, periods, underscores, or hyphens. It will surely fail if it comes across folder or file names containing spaces. At the same time, it will not detect, and therefore convert, ODF files that do not have their own default extensions (e.g., `.odt`, `.ods`, or `.odp`).

Personally, I consider these "sure failures" more of a feature than a bug for one simple reason: In my humble opinion, "limiting" yourself to files and folder names without spaces, apostrophes, and non-ASCII letters guarantees that any software or filesystem on the planet will deal with them without surprises, and it makes it much simpler to write all sorts of file-managing scripts for any purpose.

A more substantial limitation is the inability to work as intended in folders where you have no permission to create new files. Running `odfgrep` with `sudo` or giving it special SUID powers, as explained in an article online [4], would solve this problem. However, even that will not be enough to work on non-writable media like DVD archives, in which the normal `grep` tool would work just fine.

You must also take into account that `odt2txt` cannot fix stuff that "disturbs" the main text flow, like footnotes. If, for example, your ODF text contains a sentence like *Linux(1) is the best kernel around*, and *(1)* is a footnote with the text *a Unix-like kernel by Linus Torvalds*, then `odt2txt` will split the original text over five lines:

```
Linux,1

a Unix-like kernel by Linus Torvalds

is the best kernel around
```

If you were looking for the exact phrase *Linux is the best kernel around*, `odfgrep` would miss it, exactly because it was spread over multiple lines with extra text in it.

What Next?

In this tutorial you have learned why and how to use a tool that automatically scans as many ODF or text files you want, to find any given string. Cool, but why stop here?

The first thing you can do is improve `odfgrep` as you please. To work on non-writeable media, for example, you can modify it to create a temporary, complete copy of all the folders to examine in another folder. Alternatively, you can replace the test in Listing 1 (line 11) with another on the basis of the `file` command: It would be more complicated, but it would recognize ODF files no matter what their extension.

Another fun and productive line of work is using `odfgrep` as a model to build similar tools. A good candidate would be an `odfdiff` script that prints out the differences between two ODF documents.

The most important take-home lesson, however, is this: ODF is a format for sophisticated text documents, presentations, and spreadsheets that is very easy to work with and process in very efficient ways. For more proof of this, visit my little "ODF scripting" collection [5], and if you know about other scripts like those, or write new ones, please let me know! ■■■

The Author

Marco Fioretti (*http://mfioretti.com*) is a freelance author, trainer, and researcher based in Rome, Italy. He has been working with free/open source software since 1995 and on open digital standards since 2005. Marco also is a board member of the Free Knowledge Institute (*http://freeknowledge.eu*).

Info

[1] "Tutorials – Recoll" by Marco Fioretti, *Linux Pro Magazine*, issue 212, July 2018, pg. 84: *http://www.linuxpromagazine.com/Issues/2018/212/Tutorials-Recoll*

[2] odt2txt: *https://github.com/dstosberg/odt2txt*

[3] Code for this article: *ftp://linux-magazine.com/pub/listings/linux-magazine.com/213/*

[4] SUID: *www.linuxnix.com/suid-set-suid-linuxunix/*

[5] ODF scripting: *http://freesoftware.zona-m.net/tag/odf-scripting*

Odd Couple

By using two very different tools, Kdenlive and ImageMagick, you can make animation less tediousness and create some pretty cool video effects and transitions.

BY PAUL BROWN

Back in *Linux Magazine* issue 209 [1], I used FFmpeg and animation to create a transition from one video clip to another. In that instance, I used a Pacman animation with a transparent area and a blue, chroma area to make Pacman gobble up the first clip and lead in to the second.

Although that was fun, it required some serious and not too intuitive FFmpeg-fu to pull off. Also, placing the transition exactly where I needed it proved tricky, and it wasn't possible to know if I had done it right until I had rendered the final film.

That is why the Free Software gods have purveyed us with things like graphical video editors. Kdenlive [2] is currently my favorite, and, although I did say back in issue 209 that you couldn't use the chroma technique in Kdenlive in one go (you had to use one project to merge the first clip with the animation, render, then start a new project, and merge the rendered clip with the second clip), you can create the custom transition effect in one go using a different technique: masks.

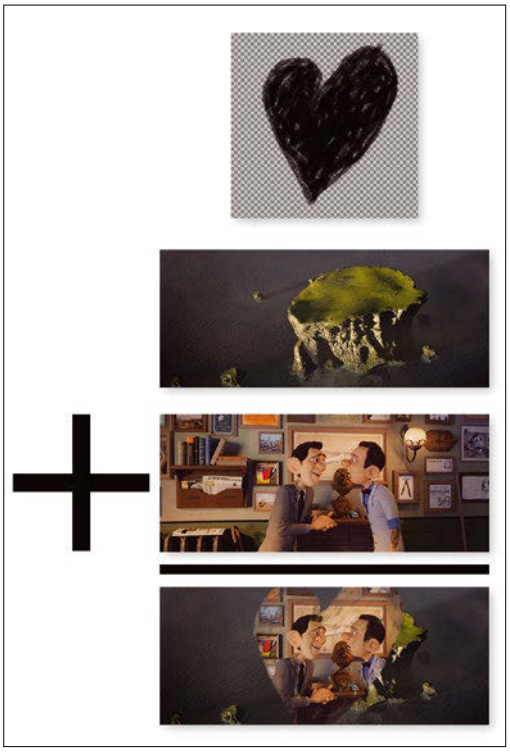


Figure 1: You can create a mask with a PNG with transparent pixels.

Kdenlive Masks

Masks in Kdenlive are just images that contain transparent and non-transparent pixels that you can combine in different ways with your clips. Fire up your favorite image editor, draw a heart shape on a transparent background, save it as a PNG, and you have a heart mask (Figure 1). That's it.

The trick is in how you apply it in Kdenlive.

Open Kdenlive and load into the project bin (first pane on the left) the two clips you want to merge, as well as the image that will serve as a mask. By default, Kdenlive gives you three video tracks to work with. You can add more, but three is perfect for your purposes. Put your image in the topmost track, the clip you want to use as a mask on the track under that, and the clip you want to show through the hole made by your mask on the bottom-most track.

To turn your image into a mask, right-click on it on the track and choose *Add transition* | *Composite and transform*. Pull the ends of the yellow transition bar so that it covers the whole length of your static image clip. With the transition still selected, click on the *Properties* tab in the *Effects and transitions* panel (second from left, next to the project bin). You will see the properties of your transition (Figure 2).

One of the last options in the properties panel is the *Compositing* drop-down. This tells Kdenlive how you want to combine your image with the clip under it. What you want is the *Xor* option. The XOR operation in this context is very simple: If a pixel in the top layer (the static image that contains opaque and transparent pixels) is colored in (i.e., is opaque), it is considered a *1*. The same applies to the pixels on the layer below containing the first clip. Most video clips don't have transparent pixels. If this is the case, all the pixels in all frames will be *1*s. As you know from your lessons in binary logic [3], a *1* XORed with another *1* gives *0*. In this transition, this means that, when an opaque pixel in the mask is overlaid onto another opaque pixel in the clip on the second track, both are rendered transparent, and the clip on the third layer will show through.

Animating

That was simple enough, but no transitioning is happening. You can make things more dynamic, because masks have keyframeable features (see the "Keyframes" box) that lets you animate them to some degree and turn your mask into a proper transition. For example, to make the second clip gradually appear in your mask and then the mask itself grow until it blots out the first clip, you can do as follows:

1. Click on your *Composite and transform* transition in the timeline and make your transition about one second long.
2. Look at the properties panel above the tracks (Figure 3), and you will notice the transition has its own timeline. Make sure there's a keyframe at the beginning of the transition – a keyframe looks like a little arrow pointing upward. A keyframe at the beginning is the default, but check just in case. Click on that keyframe (it should turn red, and all the options should become available) and set the Opacity bar to zero percent. This will make your mask completely invisible.
3. Move the transition's timeline cursor (the little triangle under the timeline) about 10 centiseconds in and double-click on the cursor. This will create a new keyframe at the 10-centisecond position. Now make your mask 100 percent opaque. This will make your mask fade in.

   If you have trouble placing the new keyframe, that may be because the cursor is not squarely on a frame. Move the cursor a little to the left or a little to the right until you find a spot that will accept a keyframe.

   Kdenlive will insert all the intermediate steps, between your two keyframes, making your mask go gradually from completely transparent and invisible, to completely opaque, showing the second clip on the bottom track.

   Next, you want to make the mask grow until it completely covers the first clip and, thus, completes the transition to the second clip.
4. Place a third keyframe about five centiseconds from the end, make sure your timeline cursor is on the keyframe (the head of the arrow will be red instead of white), and change the mask's size to cover the whole frame. To do this, you can use the dimension boxes under the timeline. Changing the values in the boxes labeled *X* (for the width of the mask) and *Y* (for its height) will change your mask's size.

   As the resizing happens from the upper left corner downward and to the right, your mask will probably be displaced as it grows. It might even be pushed completely outside the frame. To get it back to the center, go to the preview panel (the panel showing a preview of your project on the right), select the *Project Monitor* tab (Figure 4), click on the hamburger menu icon (the icon with three vertical bars to the right of the time counter), and mark the *Zoom* checkbox.

   A new toolbar will appear in the upper left-hand corner of the preview panel (in Kdenlive's development version, this bar appears when you mouse over the upper right-hand corner of the preview panel). Use the slide at the bottom of the bar to zoom out from the visible frame. At some point, you will see a red rectangle with a red square in its center. This is the frame of your mask. The visible frame also has a square in its center, and both squares are joined by a line.

   Place your cursor over the square in the center of your mask frame, click, hold, and drag the square over the visible frame's square. This will recenter your mask in the visible frame.

   Also notice that you have four handles (again in the shape of red squares) in each corner of you mask's frame. You can drag on these to change the size of your mask visually.
5. For good measure, add a *Dissolve* transition between the first and second clip (on the middle and bottom tracks), starting at the same position as the third keyframe you placed in the *Composite and transform* keyframe and ending at the same ending point of the *Composite and transform* transition. This will ensure any leftover bits from the first clip will have disappeared completely and gracefully by the end of your custom transition. Figure 5 shows how this will look on the project timeline.

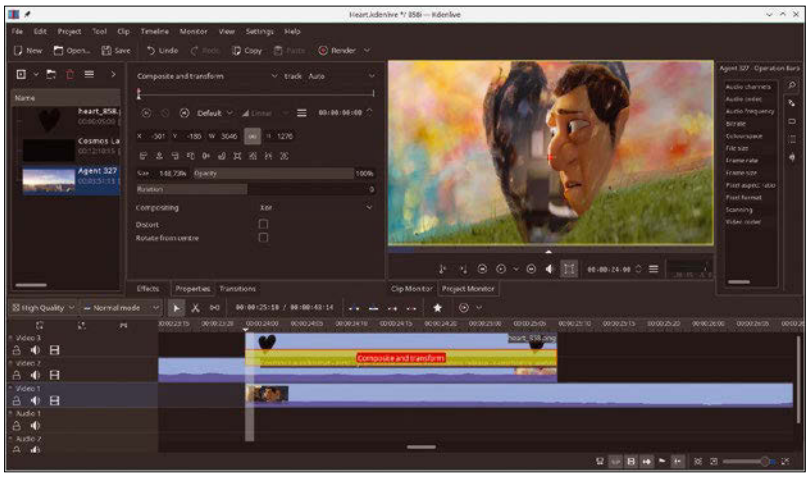


Figure 2: Set *Compositing* to *Xor* to make the pixels in the two upper layers transparent.

Cartoons

Again, this is quite straightforward. After all, you are using Kdenlive's built-in tools as they were intended. I'll mix things up a bit, because things get a little more complex when you want to make a transition with an animated mask.

The complexity comes not so much from the theory, since the principles are the same. The problem is more logistic, because now you need not just one image, but a whole sequence of PNG images. If your transition

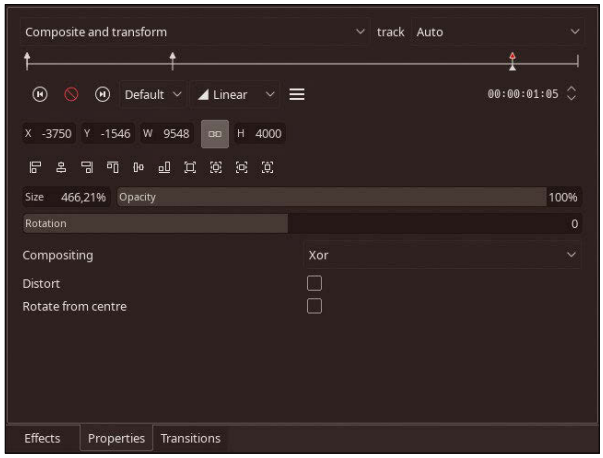


Figure 3: How to convert a simple mask into a proper transition.

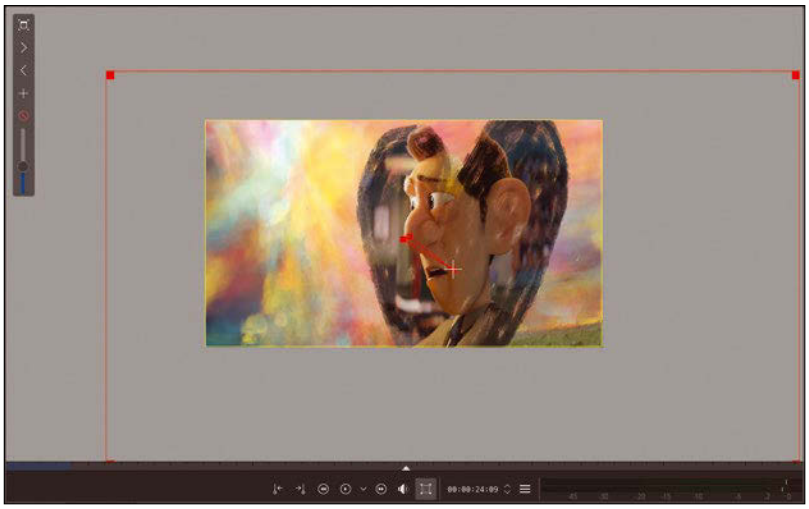


Figure 4: You can recenter your mask (in the red rectangle) by dragging its center over to the visible frame's center.

is two seconds long, that is 50 images at least. You could work on each frame by hand, one by one, but I didn't do that for the animated Pacman I used in issue 209, and I have no regrets. Instead I used ImageMagick [4].

For starters I didn't even draw the Pacman in Inkscape or anything like that. Instead, I used ImageMagick's `-draw` feature.

To generate each frame of the animated Pacman, you can use the commands shown in Listing 1. Line 1 draws Pacman with its mouth open. Here is a breakdown of how that line works:

- `convert` is ImageMagick's jack-of-all-trades command with which you can do most operations.
- `-size 534x534 xc:none` makes an image that is 534 pixels wide by 534 pixels high and sets the background to `none`, which, in a PNG image, means transparent.
- `+antialias -fill none -stroke yellow` switches anti-aliasing off (you want a 1980s pixelated vibe), sets the fill to transparent (otherwise everything you draw will be filled with the default color, black), and sets the stroke (i.e., the picture's outline) to yellow.
- `-draw 'arc 0,0 534,534 225,135'` draws Pacman's "body." The `arc` option draws, as its name implies, arcs of ellipses. To figure out your ellipse's shape, you have to imagine it enclosed in a rectangle (or square, if you want to draw a circular arc). The first set of coordinates (`0,0`) gives ImageMagick the upper left-hand corner of the rectangle surrounding the ellipse. In this case, it coincides with the upper left-hand corner of the image itself. The next set of coordinates gives ImageMagick the position of the lower right-hand corner of the rectangle. Again it coincides with the lower right-hand corner of the image. As the width and height are the same, this will be a circular arc. The last set of figures, `225,135`, draws the arc itself from 225 degrees to 135 degrees. Note that, in ImageMagick, zero degrees is on the left, 90 degrees is pointing down, 180 degrees is on the right, and 270 degrees is pointing up.
- `-draw 'line 78,78 267,267' -draw 'line 78,456 267,267'` draws the straight lines that make up Pacman's "jaws." You can figure out where to start each line with a bit of trigonometry. The endpoint in both cases is, of course, the center of Pacman's circular body.
- `-floodfill +269+269 yellow` picks a spot within Pacman's body (two pixels off from his center) and floods the whole shape with yellow.
- Finally, the resulting image is poured into `cc01.png`.

Line 2 does something similar, but draws Pacman with his mouth half closed and puts the resulting image into `cc02.png`. Finally, line 3 draws a yellow circle, which is, of course, Pacman with his mouth completely closed (Figure 6)..

Now that you have the different frames for the animation, you can create the frames for the transition. As mentioned above, if you want to make your transition last two seconds, you have to draw 50 images, putting the first frame of your animation (`cco1.png`) on the right of the frame and then, with each successive frame, move Pacman toward the left, as you cycle through each frame of the animation.

Instead of doing this by hand, you can write a Bash script (Listing 2) that uses ImageMagick instructions embedded in loops to do the repetitive work for you.

The script takes four parameters:

1. The path to the directory containing the animation frames (e.g., `animation/`)

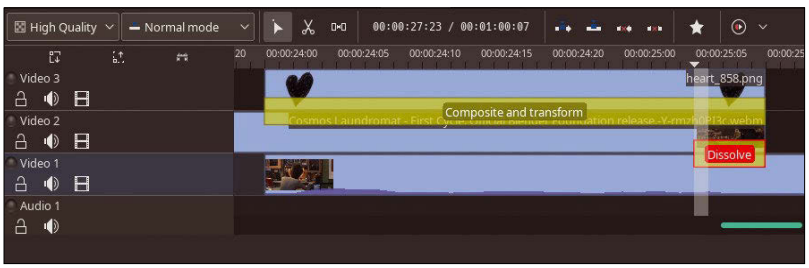


Figure 5: Add a *Dissolve* transition to clear up any remaining pixels from the first clip.

Keyframes

You use keyframes in video editing and animation to indicate when something changes and to have your application calculate the intermediate states between the changes. Say, for example, you want a mask to go from completely invisible to completely visible. You place a keyframe on the timeline where you want the mask to be transparent and set the masks opacity to zero percent. Then you place another keyframe further down the timeline, select it, and make the mask's opacity 100 percent. Kdenlive will calculate all of the intermediate levels of opacity so that the mask will gradually fade in between the two keyframes.

Listing 1: Animating Pacman

```
01 convert -size 534x534 xc:none +antialias -fill none -stroke yellow -draw 'arc 0,0 534,534 225,135'
   -draw 'line 78,78 267,267' -draw 'line 78,456 267,267' -floodfill +269+269 yellow cc01.png
02 convert -size 534x534 xc:none +antialias -fill none -stroke yellow -draw 'arc 0,0 534,534 202,158'
   -draw 'line 20,164 267,267' -draw 'line 20,369 267,267' -floodfill +269+269 yellow cc02.png
03 convert -size 534x534 xc:none +antialias -fill yellow -stroke yellow -draw 'circle 267,267 0,267' cc03.png
```

Listing 2: customtransition.sh

```
01 #!/bin/bash
02
03 mkdir -p transition/frames
04 mkdir transition/animation
05
06 f_w=`expr match "$2" '\(.[0-9]*\)'`
07 f_h=`expr match "$2" '.*x\(.[0-9]*\)'`
08
09 for i in $1*.png
10   do
11     convert $i -resize x$f_h transition/animation/${i##*/}
12 done
13
14 anim_frames=`ls $1|wc -l`
15
16 cc=(transition/animation/*.png)
17 color=$3
18 fs=$4
19
20 i_w=0
21 for i in $1/*.png
22   do
23   cfs=`convert $i -format "%w" info:`
24
25   if [ $cfs -gt $i_w ]
26     then
27       i_w=$cfs
28   fi
29 done
30
31 i_s=$[($f_w+$i_w)/$fs)+1]
32 f_l_e=$[($i_s*$fs)-$f_w]
33 p_p=$[$i_s*$fs]
34 pf_w=$[($i_s*$fs)+$i_w+1]
35
36 f_c=0
37 for i in $(seq -f "%03g" 0 $fs)
38 do
39   np=$[$p_p-(10#$i*$i_s)]
40
41   convert -size ${pf_w}x${f_h} xc:none ${cc[$f_c]}
     -geometry +${np}+0 +antialias
     -composite transition/frames/frameNF.png
42   convert transition/frames/frameNF.png -floodfill
     +$[$pf_w-1]+1 $color transition/frames/frameUC.png
43   convert transition/frames/frameNF.png -crop
     ${f_w}x${f_h}+$f_l_e+0 transition/frames/frame$i.png
44   convert transition/frames/frameUC.png -crop
     ${f_w}x${f_h}+$f_l_e+0 transition/frames/chroma$i.png
45
46   let f_c++
47   if [ $f_c -gt $((anim_frames-1)) ]
48     then
49       f_c=0
50   fi
51 done
52
53 rm transition/frames/frameNF.png
54 rm transition/frames/frameUC.png
```

2. The width and height you want for the resulting transition frames (e.g., `1280x720`)
3. Color for the chroma (e.g, "`#0f0`")
4. Number of frames you want for the transition (e.g., `50`)

An example command line would look like this:

```
customtransition.sh animation/ 1280x720 "#0f0" 50
```

From lines 3 to 34, you set up directories that the animation is going to land in and the variables you will later need to create the transition's frames. It is all about calculating the ideal sizes for the animation frames, the transition frames, and the initial positions of all the moving parts. The loop that runs from lines 9 to 12, for example, resizes the frames of your animation to make sure they are exactly the height of the transition clip.

The loop that runs from lines 21 to 29 finds the width of the widest animation frame. You do this because, before creating the final transition frames, you have to fit your animation image mostly outside the frame on the right in the first few transition frames and mostly outside the frame on the left in last few. You have to prepare for that, so the proto-frames are as wide as you

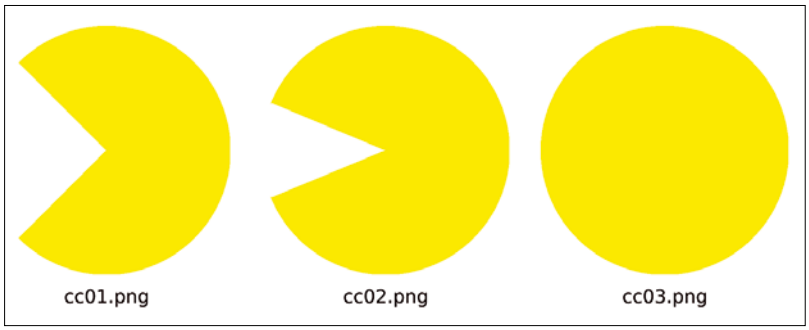


Figure 6: Use some ImageMagick-fu to create your animation frames.

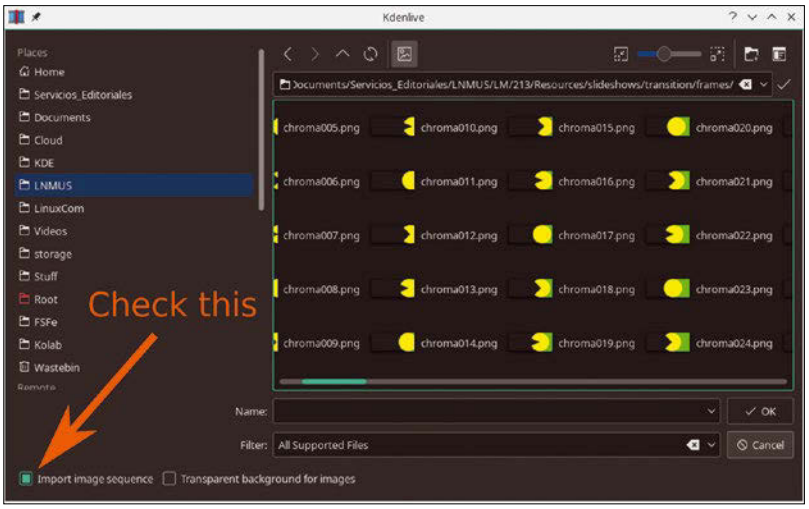


Figure 7: Mark the *Import image sequence* checkbox to import frames as a clip.

tell them to be, but also with double the width of the widest animation frame added to the width. Don't worry: They are cropped later (see lines 43 and 44).

The real work happens in the loop from lines 37 to 51. Here, you insert the animation frame in its correct position for each transition frame (line 41) and create a chroma frame (in case you need it) by filling the empty area to the right of the animation frame with the chroma color (line 42). Then you crop both transitory frames and give them their final place in the sequence (lines 43 and 44).

Lines 46 to 50 cycle through the animation frames.

Although the example of a static image (as shown in Figure 1) used a black mask, the mask doesn't have to be black. Remember that the only condition to successfully XOR the mask with the underlying clip is that the mask contain transparent and opaque pixels. The color of the opaque pixels is irrelevant. That means the chroma frames of the transition will work perfectly well as masks.

Once you run the script, you will end up with a folder (called `transition`) that contains two subfolders, `animation/` and `frames/`. The `animation/` subfolder contains your resized animation frames, but the `frames/` subfolder contains all the frames you need to create your transition in Kdenlive.

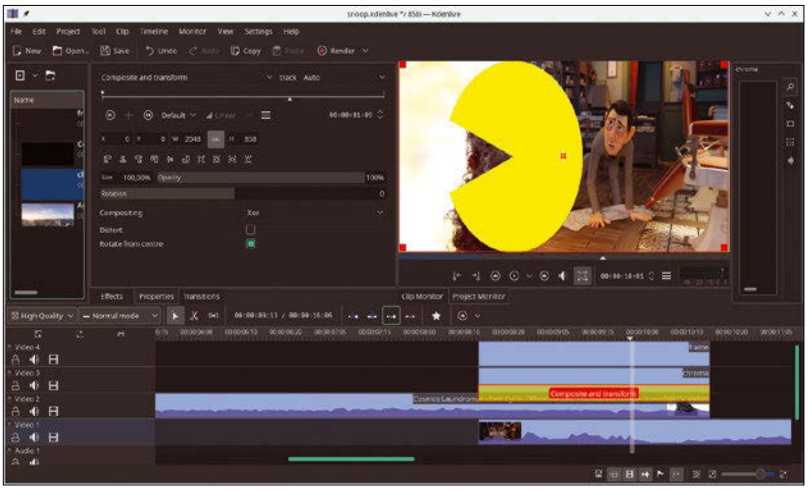


Figure 8: Align your clips with the animation and mask sequences to create a perfect transition.

To apply your customized transition, open the two clips that you want to use the transition on in your Kdenlive project; then, load the *frame* and *chroma* sequence of images that the `customtransition.sh` script built. To do that, click on the *Add clip* button in the project bin, navigate to the `transition/frames/` folder created by `customtransition.sh`, select the *Import image sequence* checkbox at the bottom of the file explorer window, and choose the first file in the *frame* sequence (`frame001.png`). Then open the *chroma* sequence by doing the same, except now pick `chroma001.png`.

Both sequences will load into Kdenlive and you will be able to work with them as if they were clips.

To set up your transition in Kdenlive, you'll need four video tracks: In the topmost track, you put the animation sequence (*frame*). On the second track, put the mask/chroma sequence (*chroma*) and align it with the animation on the top track. On the third track, put the first clip you want to transition from and align the end of that clip with the end of the animation sequence. Finally, on the bottom track, add the second clip you want to transition to, and make the beginning of that clip coincide with the beginning of the animation and chroma/mask sequences.

Add a *Composite and transform* to the chroma/mask sequence so that it affects the first clip, and change the *Compositing* method to *Xor* as shown in Figure 8.

Rendering your project will give you Pacman chomping up one scene and leading into the next.

Greening

ImageMagick can also help you adapt existing footage for your transitions.

Say you have footage of an actor dancing in front of a green screen. First, the same way Kdenlive can load a sequence of images and use it like a clip, you can do the opposite: Load a movie clip and have Kdenlive render it as a sequence of PNG images. You can then use ImageMagick to turn the green part into a transparent backdrop and then cut out the action so you can use it for your transition.

To generate a sequence of images you can later process with ImageMagick, first load the green screen footage into Kdenlive and put it onto a track in your project. Press *Render*, and, in the dialog, scroll down through the *Format* list on the *Render project* tab until you reach the *Image sequence* section. Choose PNG, because this image format has an alpha channel (i.e., transparency) built in. Choose where you want to render and the base name of the images, and click on *Render to File*. Kdenlive will split your footage into convenient images, at one image per frame.

ImageMagick has two functions that can help you get the images ready for `customtransition.sh`:

`-fill` and `-trim`. The `-fill` function can change a preexisting color in an image to another color of your choice. If you feed `-fill` the "color" `none`, it makes the pixels transparent.

Before you start, though, check the color of the green screen in one of your frames. I use KDE's *KColorChooser* for this. In the image shown in Figure 9, the green is not a pure #00ff00m, but rather #13ff09. Not all pixels are going to be the exact same green either, so you can use the `-fuzz` option to give ImageMagick a larger amount of greens to work with:

```
convert frame000.png -fuzz 40% -fill none ➥
  -opaque "#13ff09" alpha_frame000.png
```

This will change most, if not all, of the green pixels in the first frame of the sequence transparent.

ImageMagick's `-trim` function tries to crop an image automatically by cutting off monochrome parts of the background. Because all of the pixels surrounding the actor are now transparent,

```
convert alpha_frame000.png ➥
  -trim alpha_cropped_frame000.png
```

will cut off the parts surrounding the actor that are not filled with a color.

You can mush both functions into one command and put it into a loop to process a whole directory full of frames:

```
for i in frame0*.png; do convert $i ➥
  -fuzz 40% -fill none -opaque "#13ff09" ➥
  -trim alpha_cropped_${i}; done.
```

The resulting frames will work perfectly with `customtransition.sh` (Figure 10).

Shortcomings

Be aware that I wrote `customtransition.sh` for mainly educational purposes, as an example of how you imaginatively can use external tools to overcome the shortcomings of some Free Software video editors.

This means the script also has shortcomings of its own. For one, it only wipes from right to left. It would be trivial to implement transitions that ran left to right, but, for it to be a complete tool, you would probably also want top-to-bottom, bottom-to-top, center-to-edges, and edges-to-center … at the very least.

Acknowledgments

The clips used in this tutorial are taken from *Agent 327* [6] and *Cosmos Laundromat* [7], two of the great movies produced by the Blender Institute.

Figure 9: You can use basically anything you want to create a customized transition, including footage of some random dude walking across the screen.

Additionally, if there is a "gap" in the body of the animation figure – imagine, for example, an actor throwing their hat in the air – the resulting animation frames will be wrong. The script uses `floodfill` to fill in the color to the right of the actor. This means that color will flow through a gap to the left, making the whole background opaque and unusable for what you want.

That said, you can download the script [5] and try to improve it to meet your needs. Hopefully, this article will have given you enough clues on how to do that, so get creative! ■■■

Info

[1] "Creating custom transitions with FFmpeg" by Paul Brown, *Linux Magazine* issue 209, April 2018: *http://www.linux-magazine.com/Issues/2018/209/Gobbling-Up*

[2] Kdenlive: *https://kdenlive.org/en/*

[3] XOR binary logic operation: *https://en.wikipedia.org/wiki/Exclusive_or*

[4] ImageMagick: *https://www.imagemagick.org/*

[5] Download: *https://gitlab.com/customcommandlinevideoediting/customtransitions*

[6] *Agent 327*: *https://cloud.blender.org/p/agent-327/*

[7] *Cosmos Laundromat*: *https://cloud.blender.org/p/cosmos-laundromat/*

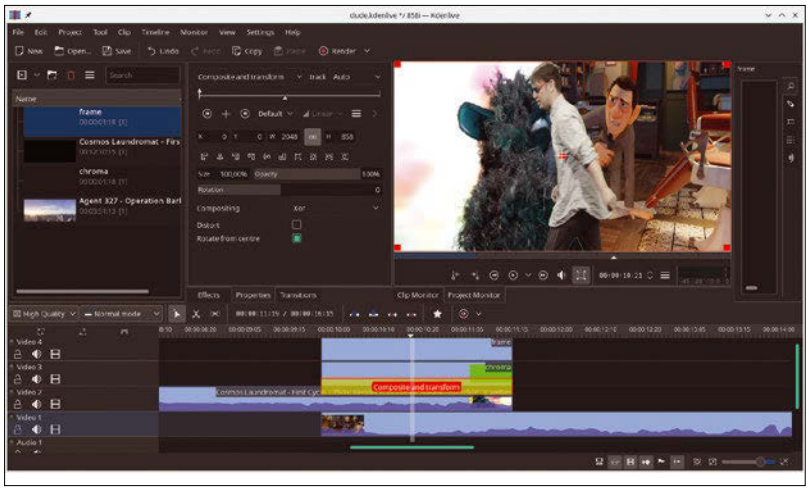


Figure 10: The final result.

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*.

Akademy 2018

Date: August 11-17, 2018

Location: Vienna, Austria

Website: *https://akademy.kde.org/2018*

Akademy 2018 is expected to draw hundreds of attendees from the global KDE Community to discuss and plan the future of the Community and its technology. Many participants from the broad free and open source software community, local organizations, and software companies also will attend.

Linux Security Summit

Date: August 27-28, 2018

Location: Vancouver, Canada

Website: *https://events.linuxfoundation.org/events/linux-security-summit-north-america-2018/*

The Linux Security Summit (LSS) is a technical forum for collaboration between Linux developers, researchers, and end users to foster community efforts in analyzing and solving Linux security challenges.

Nextcloud Conference 2018

Date: August 23-30, 2018

Location: Berlin, Germany

Website: *https://nextcloud.com/conf/*

The Nextcloud community will get together at the TU Berlin from August 23 to 30. The conference is focused on making Nextcloud better through discussions, coding, packaging, designing, translating, and more, all with the participation of our amazing community.

Events

DebConf18	July 29–August 5, 2018	Hsinchu, Taiwan	https://wiki.debconf.org/wiki/DebConf18
Akademy 2018	August 11–17, 2018	Vienna, Austria	https://akademy.kde.org/2018
Linux Security Summit	August 27–28, 2018	Vancouver, Canada	https://events.linuxfoundation.org/events/linux-security-summit-north-america-2018/
Open Source Summit North America	August 29–31, 2018	Vancouver, Canada	https://events.linuxfoundation.org/events/open-source-summit-north-america-2018/
Atlassian Summit Europe	September 3–5, 2018	Barcelona, Spain	https://www.atlassian.com/company/events/summit-europe
Drupal Europe	September 10-14, 2018	Darmstadt, Germany	https://www.drupaleurope.org/
DevOpsDays Berlin	September 12–13, 2018	Berlin, Germany	https://www.devopsdays.org/events/2018-berlin/welcome/
The Linux Foundation Legal Summit	September 12–14, 2018	San Francisco, California	https://events.linuxfoundation.org/events/lf-member-legal-summit-2018/
Open Source Firmware Conf.	September 12–15, 2018	Erlangen, Germany	https://osfc.io/
Storage Developer Conf.	September 24–27, 2018	Santa Clara, California	https://www.snia.org/events/storage-developer
Open Networking Summit Europe	September 25–27, 2018	Amsterdam, Netherlands	https://events.linuxfoundation.org/events/open-networking-summit-europe-2018/
Open Source Backup Conference 2018	September 26–27, 2018	Cologne, Germany	https://upcoming.org/event/open-source-backup-conference-2018-0xlw0zogn5
All Systems Go	September 28-30, 2018	Berlin, Germany	https://all-systems-go.io/
All Things Open	October 21-23, 2018	Raleigh, North Carolina	https://allthingsopen.org/
Open Source Summit Europe	October 22–24, 2018	Edinburgh, UK	http://events.linuxfoundation.org/events/open-source-summit-europe
Linux Presentation Day 2018.2	November 10, 2018	Europe-wide in many cities	http://www.linux-presentation-day.de/
Linux Kernel Summit	November 12–15, 2018	Vancouver, Canada	https://events.linuxfoundation.org/events/linux-kernel-summit-2018/
IT-Days 2018	December 10-13, 2018	Frankfurt, Germany	https://www.ittage.informatik-aktuell.de/

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

Contact Info

Editor in Chief
Joe Casad, jcasad@linux-magazine.com

Copy Editor
Amy Pettle

News Editor
Swapnil Bhartiya

Editor Emerita Nomadica
Rita L Sooby

Localization & Translation
Ian Travis

Layout
Dena Friesen, Lori White

Cover Design
Lori White

Cover Image
© Sergey Mironov, 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

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.

Issue 214 / September 2018

Systemd

Many Linux users are still getting acquainted with the tools and tips for managing the systemd init daemon. Next month we delve into systemd.

Approximate On Sale Date

UK / Europe	Aug 04
USA / Canada	Aug 31
Australia	Oct 01

Preview Newsletter

The Linux Magazine Preview is a monthly email newsletter that gives you a sneak peek at the next issue, including links to articles posted online.

Sign up at: ***www.linux-magazine.com/newsletter***

Photo by mauro paillex on Unsplash

Complete Your Open Source Library with Archive DVDs!

Each fully-searchable archive DVD includes past articles so you can find the content you need quickly.

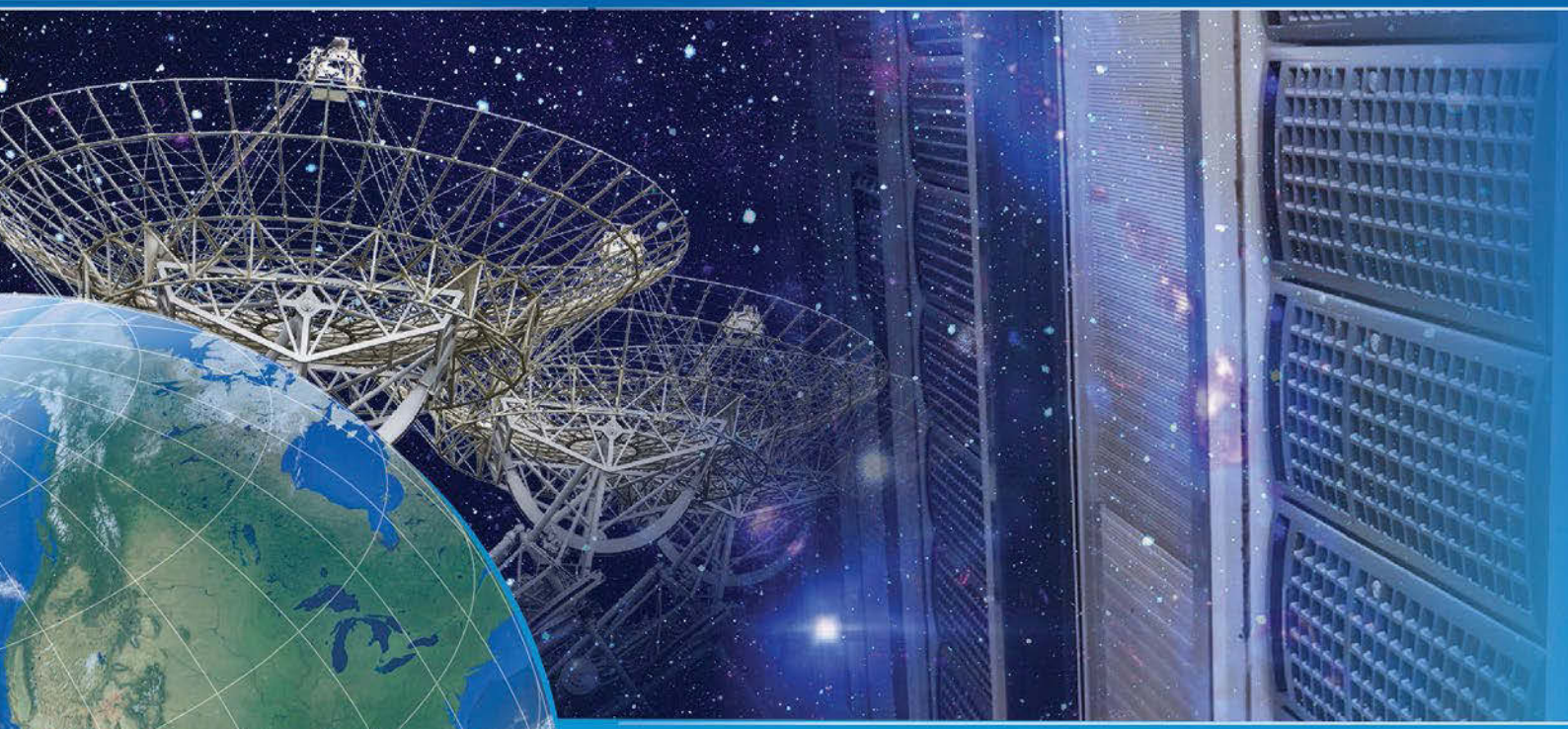
Save hundreds off the print and digital copy rate with a convenient archive DVD!

Order Your DVD Now!

shop.linuxnewmedia.com

Application-Optimized Server Solutions

Scientific Computing Built to Meet Your Specific Needs



The Fastest Image Processor in the World

With 41 of our GPU optimized SYS-7048GR-TR servers, the New GPU based Supercomputer named ARTS (Apertif Radio Transient System) is now the Fastest in the Netherlands with a calculation capacity of two PetaFLOPS. This supercomputer is Application-Optimized to scan 20,000 images per second then uses machine learning algorithms to detect if there are any fast explosions in space using one of the largest Radio Telescopes in the World. For Exactly the Best Enterprise Computing Solution for your Application, Choose Supermicro.

intel XEON PLATINUM inside™

Intel Inside®. Powerful Productivity Outside.

SUPERMICRO®

Application-Optimized Server Solutions

Learn more at supermicro.com/application-optimized