FREE DVD

archlinux
64-bit

ISSUE 211   JUN 2018

LINUX
18.3 "MATE" 64-bit
Linux Mint
DVD

ISSUE 211   JUN 2018
LINUX
MAGAZINE

DVD

Double-Sided DVD
INSIDE!

PIRATEBOX
Handy anonymous file server
for parties and meetups

MYCROFT
An open source
personal assistant

LINUX PRO
MAGAZINE

JUNE 2018

MYCROFT

Can an open source personal assistant compete with Alexa?

Scanners and Linux

Peppermint OS
Peppy distro built for the cloud

Meltdown & Spectre
Kernel developers take
swift action to protect Linux

Knight's Tour
Classic chess puzzle gets a Python twist

Node-RED
Control a Rasp Pi
using text messages

LINUXVOICE

- maddog: A brief history of *x
- Resetter: Reset your broken system
- Audacity: Create a podcast
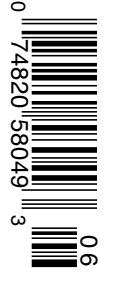- Gravit Designer: Is this alternative
  vector graphics tool ready?

FOSSPicks
- Krita 4.0
- FreeTube
- OpenSnitch
- Yoda

Tutorial
Create a desktop
app with QML

Issue 211
June 2018
US$ 15.99
CAN$ 17.99

0 74820 58049 3
06

ROOT OF TRUST

Dear Reader,

Every day, things seem to happen that I never thought I'd see, but one recent announcement was particularly striking if you've been around the FOSS community anytime in the last 20 years: Microsoft just proclaimed they are creating their own custom version of the Linux kernel. A Linux kernel from Microsoft is big news if you remember the old times, when they used to say Linux was a "virus," and they used all their monopoly power in an attempt to squelch and obliterate the Linux menace.

As I write this column, the curious news is trickling out onto the high-tech blogs and news sites. Most of the stories paraphrase or quote from the original announcement in a blog at the Microsoft Azure website [1]. What they are really announcing is a new IoT system with three components:

- the cloud-based Azure Sphere Security Service
- a new class of Azure-certified microcontrollers (MCUs) to go in IoT devices
- Azure Sphere OS, which is actually the Linux kernel with Microsoft modifications

They needed a new operating system for their new IoT environment, and instead of choosing Windows IoT or some spin of their embedded Windows OS, they chose to use Linux instead. Does that mean they admit Linux is better? Not out loud at least: They say Windows IoT is too big to run at the small scale they need for the Azure Sphere environment. (Hmmm … does that seem real, or do you think maybe they think Linux is better?) In any case, Linux is obviously better at scaling down to the size they need.

Is Microsoft part of the team now? Well, before you uncork the champagne, better to look a little deeper. The Azure Sphere program isn't really about selling software; Microsoft's profit model appears to center around the cloud service and the "Azure Sphere Certified" IoT devices. That shouldn't be a surprise to anyone. (If your profit model depends on selling Linux as a software product, you're in a lot of trouble, since most people are giving it away).

But Microsoft has lots of other ways to make money in the Azure Sphere. Cloud services, certification for hardware, consulting … all this does sound a little more like the open source environment, where the software is free and revenue comes from the surrounding services. Open source means open, right? Can anyone get involved with this promising new market?

That's where you have to remember with whom you're dealing. A closer look at the announcement offers a more nuanced view of this brave new space. The Azure Sphere announcement is strongly focused on the topic of security. The term "secure" and "security" appear a total of 35 times in the single blog post announcing the new initiative. It is obvious they plan to use security as a way to help distinguish themselves from other IoT platforms. To find out what they mean by "security," click the link in the announcement that goes to another page entitled "Seven Properties of Highly Secure Devices" [2].

Leading off the list of the properties for secure devices is something they call "Hardware Based Root of Trust." This "root of trust" refers to the work of the Trusted Computing Group (TCG), a consortium started by Microsoft and some hardware vendors several years ago that now consists of around 100 companies. TCG has a standard for the Trusted Platform Module (TPM), which is now included with many computer hardware systems. The TPM theoretically gives the hardware vendor complete control over what software can run on the system.

The result of the TCG controls is that you could have a completely free operating system running in a completely closed, vendor lock-in style computing environment. As a result, TCG and the "Trusted Computing" paradigm has come under fire from many corners of the Free Software community – most colorfully, perhaps, from Free Software Foundation founder Richard Stallman, who refers to "trusted computing" as "treacherous computing" [3].

The fact that Microsoft lists "root of trust" as the first property on their "Seven Properties of Highly Secured Devices" gives the strong impression that they intend to employ TCG technologies to maintain tight control over what software runs in their Azure Sphere IoT system. If so, the question is: Is Azure Sphere an example of Microsoft getting to be more like Linux? Or are they just getting Linux to be more like Microsoft?

Joe

Joe Casad,
Editor in Chief

Info

[1] Introducing Microsoft Azure Sphere: Secure and Power the Intelligent Edge: *https://azure.microsoft.com/en-us/blog/introducing-microsoft-azure-sphere-secure-and-power-the-intelligent-edge/*

[2] Seven Properties of Highly Secure Devices: *https://www.microsoft.com/en-us/research/wp-content/uploads/2017/03/SevenPropertiesofHighlySecureDevices.pdf*

[3] Can You Trust Your Computer?: *https://www.gnu.org/philosophy/can-you-trust.en.html*

WHAT'S INSIDE

You can't get more sci fi than a little box that listens to your instructions and talks back. Most of the leading personal assistants come in very opaque little boxes, but the open source community is hard at work on finding its own solutions. Mycroft is an open source assistant that comes on a hardware device, but you can also download the software and install it on a Linux system.

Other highlights:

- **Meltdown and Spectre** – These catastrophic hardware bugs have created problems for computer users around world. Find out what Linux developers are doing to clean up the mess (page 36).

- **Knight's Tour** – Practice your skills with a Python solution to this classic chess puzzle (page 42).

Check out MakerSpace for a look at how to control your Raspberry Pi from an Android, and visit LinuxVoice for a tutorial on writing apps with the Qt Modeling Language (QML).

SERVICE

NEWS

REVIEWS

COVER STORIES

TWO TERRIFIC **DISTROS**
DOUBLE-SIDED DVD!

LINUXVOICE

On the DVD

Linux Mint 18.3 "Mate" (64-bit)

Linux Mint targets the everyday user and puts the emphasis on an easy desktop experience. A pleasing combination of seamless operation and good design, along with a practical line-up of user-friendly applications, has kept Mint in the top spot on the DistroWatch page hit list for years. The Mate edition features the popular and practical Mate desktop, a modern continuation of the classic Gnome 2 code base. Highlights for version 18.3 include an improved Software Manager, expanded backup capabilities, and a more-configurable login screen.

Arch Linux (64-bit)

Arch Linux is a lean and powerful independent Linux with a loyal community of users who prefer Arch's close-to-the-system clarity over the cumbersome niceties of a GUI-based desktop Linux. The developers focus on "simplicity, minimalism, and code elegance." A bare-bones base system lets the user build and fine tune the operating environment with minimal bloat. Arch uses a rolling release system that allows it to provide "the latest stable versions of most software."

TWO TERRIFIC DISTROS

DOUBLE-SIDED DVD!
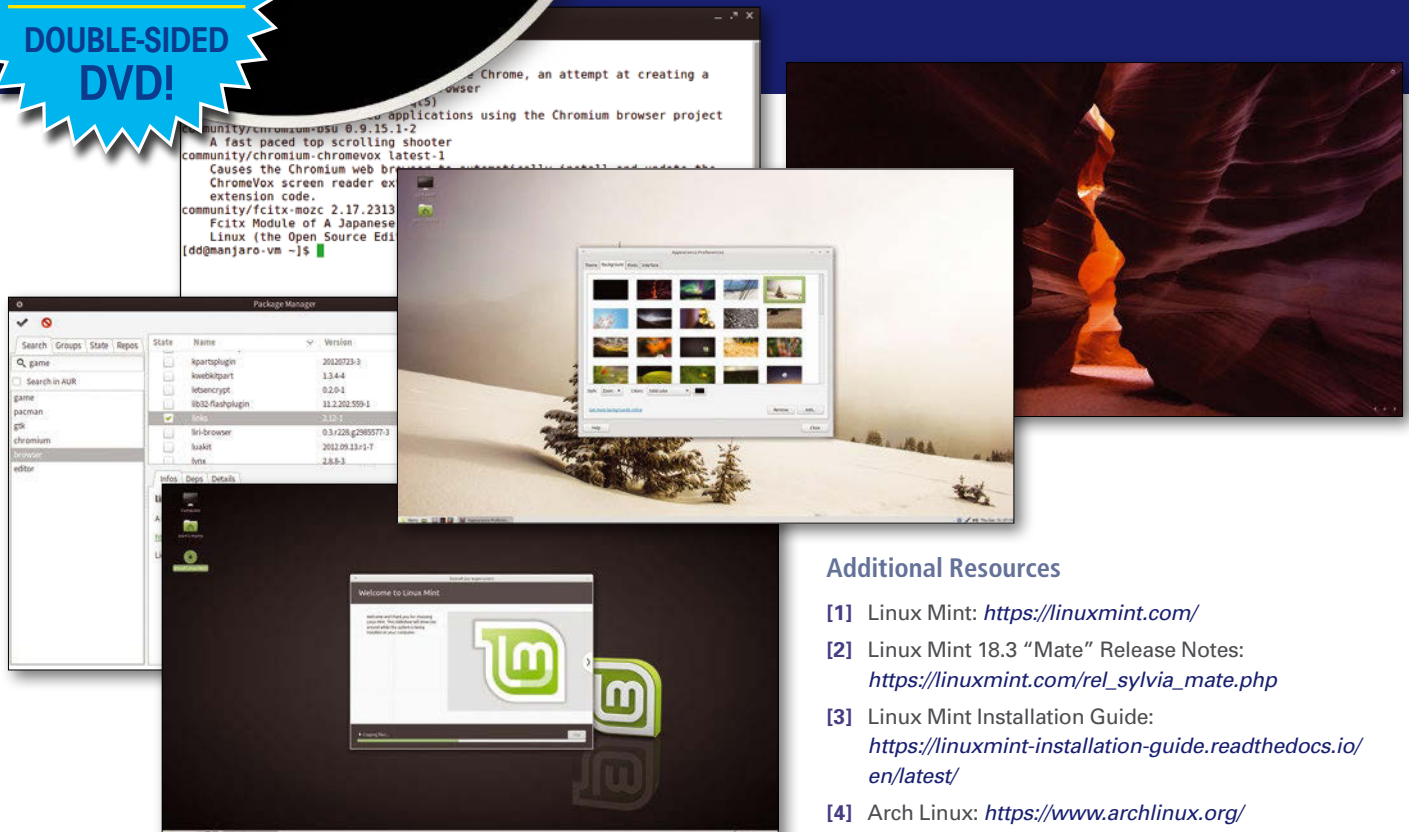
Additional Resources

[1] Linux Mint: *https://linuxmint.com/*

[2] Linux Mint 18.3 "Mate" Release Notes: *https://linuxmint.com/rel_sylvia_mate.php*

[3] Linux Mint Installation Guide: *https://linuxmint-installation-guide.readthedocs.io/en/latest/*

[4] Arch Linux: *https://www.archlinux.org/*

[5] Arch Linux Forums: *https://bbs.archlinux.org/*

[6] Arch Linux Wiki: *https://wiki.archlinux.org/*

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

NEWS

Updates on tech──────────────────────────────s
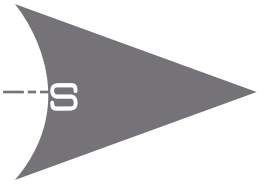
Solomon Hykes Leaves Daily Operations at Docker

Docker founder Solomon Hykes has announced that he is leaving the company's daily operations. Docker started in 2013, when Hykes rebranded dotCloud as Docker Inc. and appointed a new CEO with the goal of successfully monetizing the power of container technology. The company celebrated its fifth anniversary in March. In the five years since, Docker has pioneered containerization. However, as containers and orchestration became commoditized, Docker Inc. has had to explore other ways to look for revenue.

According to his announcement, Hykes sees a major growth opportunity with the Docker Enterprise Edition and feels the company needs to find a CTO "...with decades of experience shipping and supporting software for the largest corporations in the world." Hykes has been evolving his role within Docker for a while. Last year he stepped down from the role of CTO and became Chief Architect and Vice-Chairman of the Board of Directors. He had already moved out of the day to day involvement with the project and designated most of the work to maintainers.

Hykes wrote in blog post, "A founder's departure is usually seen as a dramatic event. Sadly, I must report that reality is far less exciting in this case. I've had many roles at Docker over the years, and today I have a new, final one – as an active board member, a major shareholder and, I expect, a high maintenance Docker user. But I will no longer be part of day-to-day operations."
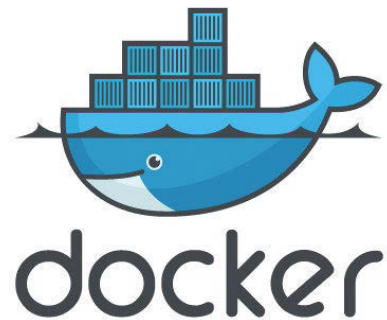
Red Hat Celebrates 25th Anniversary with a New Code Portal

Founded in 1993, two years after Linus Torvalds announced the Linux kernel, Red Hat has just celebrated its 25th anniversary in March 2018.

Red Hat was cofounded by Bob Young as ACC Corporation to sell Linux and Unix accessories. One year later, Marc Ewing created a Linux distribution called Red Hat Linux. Later, Young acquired Ewing's business and created what we know as Red Hat today.

Red Hat pioneered a business model around Linux and open source as it moved away from selling coffee mugs and merchandise and started a subscription-based business model. The company went public in 1999 and, fast forward to 2018, Red Hat reported revenue of ~ $3 billion in 2017.

Celebrating its anniversary in a pure open source manner, Red Hat announced a brand new GitHub page to host the source code of all of its projects (*https://redhatofficial.github.io/#!/main*). "The page will try to list every known free and open source project hosted on GitHub in which Red Hat staffers directly participate as part of their work," wrote Red Hat community analyst Brian Proffitt in a blog post.

Gnome 3.28 Released

The Gnome Foundation has announced the release of Gnome 3.28 "Chongqing" to honor the Gnome.Asia 2017 team. Gnome strives for simplicity to make it easier for a user to use desktop Linux. The Gnome desktop uses simple names for applications – Files, Web, Software, Videos – which accompany a simple user interface. Gnome 3.28 continues that trend.

According to the release notes, GNOME 3.28 offers two important new features that make it easier to keep track of the things. You can now star files and folders in Files (Gnome's file manager). Once they've been added, starred items can be easily viewed in a special location that can be opened from the sidebar. That's a handy feature to get quick access to files that you want.

The second feature is the arrival of "favorites" to the Contact app. Now users can pin often-used contacts, similar to iOS and Android devices. The Contact app also allows users to sort by first and last names.

The Boxes virtual machine manager offers a number of enhancements, including a new feature that makes passing files between host and guest systems easier.

There have been notable improvements in the entertainment department. The Photos app can now import files from removable devices, such as SD cards and USB drives. According to the release notes, this feature automatically detects devices that contain new images, and it also allows organizing new images into albums as they're imported.

With 3.28, Gnome has introduced a new app called Usage that tells you which app is consuming what system resources. Since the tool is just being introduced, it's very basic and includes features for examining CPU and memory consumption. It highlights problem areas, so you can easily identify issues and solve them.

From a security point of view, this release comes with extended device support. The integrated Thunderbolt 3 connection support offers security checks to prevent data theft through unauthorized Thunderbolt 3 connections.

The list of new features is long (*https://help.gnome.org/misc/release-notes/3.28/*), and the best way to learn about them is by updating Gnome on your system.

Install Firefox in a Snap on Linux

Linux desktop has an app fragmentation problem. Each distribution has its own application distribution mechanism, which ends up duplicating maintainer resources and is almost always a bottleneck when it comes to delivering updates to apps.

Linux desktop communities are trying to solve that problem with solutions like AppImage, Flatpak, and Snaps. While Flatpak is backed by Red Hat/Fedora developers, Snaps is backed by Canonical. AppImage is relatively independent. Once again there is fragmentation, meaning either app developers "waste" developer resources and create a package for all three formats or choose one. Eventually the Linux world may settle down on one, but for now we have to deal with all three.

Mozilla has officially picked Snap to offer Firefox browser for Linux. According to Canonical, by launching as it a snap, the Firefox Quantum browser is available to

an increased number of Linux users with the snap working natively on Ubuntu, Arch, Linux Mint, Fedora, Solus, Debian, and other Linux distributions that support snaps.

"Mozilla has long been a leader in the open source space," said Jamie Bennett, VP of Engineering, Devices, and IoT at Canonical. "As such, we are very happy to announce that they are joining the community of applications already available as snaps. Through their unique format, snaps can help bring some of the world's most popular apps to almost any Linux desktop, server, device or cloud machine, allowing users to select the right distro for them without having to worry about updates, security or compatibility issues further down the line."

There are a lot of advantages of using a Snap-like mechanism over the traditional method as you get updates as soon as they are released; there is no need to add third-party repositories or wait for weeks for official packages to land in official repositories.

If you want to grab a snap of Firefox, visit this link: *https://snapcraft.io/store.*

Arduino Adds Rasp Pi and BeagleBone to the Arduino Create Platform

At the Embedded Linux Conference & OpenIoT Summit, Arduino has announced support for new architectures for its Arduino Create platform for the development of IoT applications.

In an interview, Massimo Banzi, the cofounder of Arduino told me that they are looking at IoT as one of the most potential use cases and want to help the community in building projects targeting IoT. Arduino has also built a cloud, using Kubernetes and AWS to enable developers to leverage the device and cloud sides of the IoT spectrum. This announcement fits perfectly with that strategy.

Thanks to the Arduino Create platform, Arduino users can manage and program a wide range of popular Linux single-board computers like the AAEON UP board, Raspberry Pi, and Beagle-Bone as regular Arduino boards.

"With this release, Arduino extends its reach into edge computing, enabling anybody with Arduino programming experience to manage and develop complex multi-architecture IoT applications on gateways," said Banzi. "This is an important step forward in democratizing access to the professional Internet of Things."

In a blog post, the project said that multiple Arduino programs could run simultaneously on a Linux-based board and interact and communicate with each other, leveraging the capabilities provided by the new Arduino Connector. Moreover, IoT devices can be managed and updated remotely, independently from where they are located.

Source: *https://blog.arduino.cc/2018/03/13/you-can-now-use-arduino-to-program-linux-iot-devices/*

IBM Launches a New Data Science Platform

IBM has announced the launch of a new data science and machine learning platform. The new IBM Cloud Private for Data is an "integrated data science, data engineering, and app building platform." The purpose of the new platform is to let users "...build and exploit event-driven applications capable of analyzing the torrents of data from things like IoT sensors, online commerce, mobile devices, etc."

According to IBM, Cloud Private for Data is an application that operates above the Kubernetes container system, which means a user could deploy it in minutes. The solution, which includes key capabilities from IBM's Data Science Experience and Information Analyzer, is designed to help clients discover insights from their core business data.

Zack's Kernel News

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

By Zack Brown

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**.

Improving Netfilter Efficiency

Netfilter has some speed issues. Speed is always a focus of Linux development, but recent workarounds for widespread Intel hardware security flaws have resulted in significant slowdowns in the kernel. So lately, there's been even more incentive to improve speed wherever possible.

Netfilter is a generic kernel tool that allows system administrators to perform a wide array of operations on data packets moving through a network. However, as Imre Palik pointed out recently, netfilter was implemented with flexibility in mind, rather than efficiency. Even when a system performs no operations at all on network packets, simply hitting the netfilter hooks can slow things down a lot.

Imre posted a patch to address this issue. His idea was that if netfilter wasn't being used, then the kernel shouldn't hit its code at all. This would eliminate the slowdown. Of course, for systems that did use netfilter, the slowdown would remain. And this proved to be the big stumbling block for his patch.

Originally he reported a 15 percent speedup when using his patch. That would be enough to get anyone's attention, except for the fact that, as David Miller pointed out, Imre's measurements were not done with the standard `perf` tool; they were just his own observations, and thus could not be verified or even well understood. As one of the main gatekeepers of networking code in the kernel, David did not want to apply patches that were too speculative.

But David also objected to the whole idea behind Imre's patch. The solution to netfilter being too slow, he felt, was not to bypass netfilter in the case where it wasn't needed. The solution was to speed up netfilter so that it would run faster for everybody. He said, "I definitely would rather see the fundamental issue addressed rather than poking at it randomly with knobs for this case and that."

An interesting aspect of the whole discussion is the effort of a kernel "lieutenant" to guide developers towards working on a problem that might be more difficult than what they had originally attempted, but that would ultimately provide a better solution in general. There's always a little push and pull at this level, because sometimes the thing that really needs to be worked on is a really tough nut to crack, while there might be significant benefits to plucking whatever low-hanging fruit might be available. At what point does the lieutenant or the feature maintainer accept that an improvement is still good, even if it doesn't solve the big problem?

Protecting Memory from Malicious Modification

Sometimes patches slide so smoothly into the kernel, it almost doesn't seem as though anyone worked on it at all. For example, patches that don't touch anything in the "fast path" and therefore won't have a critical impact on the speed of the whole system, or patches that enhance an existing feature in a simple and obvious way, or patches that implement an optional security feature.

Igor Stoppa had an idea for an optional security feature and recently posted a patch to implement it. He wanted a way to set a memory pool dynamically to be read only. This would let the user do all the work of setting up a batch of data structures and then freeze them so that no hostile attackers could modify them on the running system.

Memory "pools" are just groups of memory allocations that are all the

same size. Linux groups them together this way because it helps make allocations take the same amount of time to set up, which in turn makes the performance of any given piece of code more predictable.

With Igor's code, the user could perform a one-time, non-reversible operation to make all the allocations in a given memory pool read only. He felt it was important that the operation be non-reversible, because if the user were able to reverse it, an attacker potentially could as well.

There was immediate interest in this patch, particularly from the XFS people. Dave Chinner remarked that XFS had a lot of data that was simply static and unchanging; so it would be a short jump to just lock it down securely with Igor's patch. In fact, a bunch of people had been discussing exactly this feature at a recent Linux conference, so it was ready to be welcomed with open arms.

Several folks immediately began discussing possible improvements, naming conventions, and other elements of the patch. But everyone seemed to think it was a winner and should go into the tree.

It's fun to read this kind of discussion on the Linux Kernel Mailing List, because it could have just as easily been the case that Igor's patch would have touched a piece of untouchable code, or mangled something that shouldn't be mangled, or slowed something down that should be fast. You never know who's going to pop out of some distant part of the kernel infrastructure, to say, "Hey! Your code is stomping my feature!" So it must be very satisfying for developers when they write a patch, and it just snaps right in.

Speeding Up Workarounds for Intel Security Flaws

Intel's Meltdown flaw has really been an annoyance to Linux developers. The primary workaround is page table isolation (PTI), a heavy-handed patch that makes the system a lot slower but that definitely handles the problem case. Before PTI, there was no obvious way to deal with Meltdown. With PTI in place, the security issues have been eliminated, which has bought time to think about how to replace PTI piece-

meal throughout the kernel, with solutions that don't have as big of a performance hit.

It's essentially the same approach that they ultimately took to eliminate the big kernel lock (BKL) some years back. When they tried to get rid of it all at once, they couldn't do it, and there was much gnashing of teeth and clawing of one another's eyes. But when they finally isolated all the BKL uses so that each occurrence could be dealt with individually, they were able to begin the gradual process of replacing them one by one with more targeted and efficient locking code.

That experience is possibly what made PTI such an obviously good choice in spite of its speed impact. Like the BKL, it solves the crucial problem, and like the BKL, there's no other apparent solution that can work everywhere in the kernel. But now like the BKL, PTI can be isolated, and each part of the kernel eventually purged of it, replacing PTI with whatever works best at that particular spot.

A recent patch from Nadav Amit tried to replace PTI at one spot in the kernel. The resulting discussion helped clarify the sort of issues the developers have to consider and why it might be much more difficult to eliminate PTI than it was to eliminate the BKL.

Nadav's idea was that Linux's compatibility mode had some of its own natural safeguards against Meltdown and that the remaining problems within compatibility mode could be handled individually, thus eliminating the need for PTI.

Compatibility mode is Linux's way of emulating older CPUs to support older user software on modern systems. In particular, some of the emulated systems are 32-bit processors, which were not susceptible to the Meltdown bug. Nadav wanted his code to detect when 32-bit user code was running in compatibility mode and to ditch PTI for that activity.

It was essentially a corner-case solution, since it wouldn't help users running regular programs on 64-bit systems, which is the much more common case. However, it seemed like a legitimate inroad against the need for PTI.

But there were problems. First of all, as Andrew Cooper pointed out, it was possible for 32-bit code to break out of

compatibility mode, get back into a 64-bit environment, and then take advantage of the Meltdown bug after all. As he put it, "Being 32 bit is itself sufficient protection against Meltdown (as long as there [is] nothing interesting of the kernels mapped below the 4G boundary). However, a 32-bit compatibility process [tries] to attack with Spectre/SP2 to redirect speculation back into user space, at which point (if successful) the pipeline will be speculating in 64-bit mode, and Meltdown is back on the table."

However, he also went on to say that Supervisor Mode Execution Protection (SMEP) could also guard against the problem he'd just identified. SMEP is a CPU feature that can prevent all non-root-level code from running. Using SMEP in short bursts is a useful way to ensure that a particular stretch of code will not be vulnerable to attack. As Dave Hansen put it, "SMEP is valuable. It's valuable to everything, compatibility-mode or not."

But there was further criticism of Nadav's patch. Andy Lutomirski pointed out that compatibility mode was not a true hardware "mode" like SMEP, but was just a form of software emulation implemented by Linux itself. As a result, he said, he wanted to discourage Nadav and others from trying to distinguish between software running in compatibility mode or not in compatibility mode. He said there wasn't really any such thing; it was all just code running in Linux. As such, Andy said, trying to identify a test for when something was in or out of compatibility mode would be complex and prone to error.

A better approach, he said, though not without other problems, would be for the user software itself to put itself into a 32-bit execution environment. Once that happened, Nadav's code could swing into action. This would eliminate any need to guess when someone was in compatibility mode, though as Andy himself acknowledged, it would require old user software to be updated. That's not always possible, for example, if only a binary executable is available. And Linus Torvalds has traditionally been adamant that old user code should continue to work under Linux unless absolutely necessary.

Linus also had his own reservations about Nadav's patch, as well as about the general prospects of eliminating PTI in Linux anywhere else. Speaking only about Nadav's patch, though, Linus said that it seemed too easy for user code to escape compatibility mode and get back to a 64-bit environment; in which case, disabling PTI would be opening the door for an attacker.

Nadav replied that his code attempted to cover nearly all cases where user code might try to break out of compatibility mode, though he acknowledged, "There is one corner case I did not cover (LAR), and Andy felt this scheme is too complicated. Unfortunately, I don't have a better scheme in mind."

But even covering all the cases, Linus said, was "some really shady stuff" that might be too difficult to maintain safely. He said, "if you get it wrong, things will happily work, except you've now defeated PTI. But you'll never notice, because you won't be testing for it, and the only people who will are the black hats." Linus said that the fragility of trying to cover all these cases, "makes me go 'eww' about the whole model. Get one thing wrong, and you'll blow all the PTI code out of the water."

Linus also pointed out that Nadav's whole patch was covering a small corner case that wouldn't make any difference to most users, but that could potentially be used by malicious attackers to defeat PTI across the entire system. At one point Linus remarked, "I just feel this all is a nightmare."

So it's not quite as simple as getting rid of the BKL. And getting rid of the BKL wasn't simple by any stretch of the imagination. But in the case of PTI, the risks are greater. Get one thing wrong among a complex set of safeguards in any part of the kernel, and the whole system is vulnerable to attack. Meanwhile, the reason PTI had to be so heavy-handed in the first place is exactly because any other approach will have to be complex, difficult, and error prone. The decision to have a near-complete separation between kernel and user page tables is simple and solves the problem. Re-exposing those kernel page tables to user space will only be possible under very delicate and context-sensitive conditions. ∎∎∎

KDE looks at the path ahead

What's Next?

Two leaders of the KDE project report on priorities and celebrate a funding victory. *By Swapnil Bhartiya*

The KDE project recently received a $200,000 donation from the Pineapple Fund. Swap sat down with Lydia Pintscher, the president of KDE e.V. board, and Sebastian Kügler, the former vice president of the board, to understand the goals of the KDE project and how the KDE community will use the funds to further those goals.

KDE is one of the oldest open source communities. The KDE project was founded by Matthias Ettrich in 1996 as an attempt to build a better desktop for Unix systems. Today, the KDE project has evolved into a massive community that creates more than just a desktop environment for Linux and BSD. To better manage a growing community, Ettrich also created a non-profit organization called KDE e.V.

KDE e.V. supports the KDE community in legal and financial matters. According to Lydia Pintscher, "We also fund KDE-related events like Akademy and developer sprints. We pay the bills for the servers that host the source code, in addition to a lot of other things."

Most KDE financial support comes in one of three ways. Pintscher explained, "The one that's closest to my heart is individual contributions. People donate anything from fl2 to fl100 because they like our software and want to support it."

Another way KDE e.V. receives funding is through membership fees for patrons. "It's for companies that value our code and use it. They want to support it financially." The third way, is through sponsorship of Akademy, an annual event for KDE contributors.

Areas of Focus

Currently, the KDE community has identified two important priorities. "One is a bit conservative angle; we are trying to make the software more stable and polishing it," said Kügler. "We are doing really well, and our public perception is getting better. Users really seem to appreciate the strategy of polishing the current codebase, making it slicker and fixing bugs. Now we can polish without large architectural changes. We have a focus on what really matters to users and what gets us positive feedback: stability and performance."

The other critical concern is privacy. "More than ever, what this world really needs is tools to protect privacy. The tools that give you the possibility of leading a private life," he said, "I think identity theft is going to become a really huge problem in the future."

We have already seen cases where both private companies and government

Lydia Pintscher Image By VGrigas (WMF) – Own work, CC BY-SA 3

agencies have had their systems compromised. As we are becoming more dependent on software, cyberwarfare is going to heat up. Attackers can shut down power supplies, or even cause damage to civil infrastructure. "We are building tools and rallying the community to build strong software and help people fend against these things," he said.

Pintscher and Kügler point out that the market for the PC itself is declining. Instead of PCs, people are relying more on embedded hardware and mobile devices. That's exactly where the KDE community is making future investments. "We are creating the Plasma Mobile platform. Not much of it is visible right now, but we are working on making it run really well on smaller embedded devices."

The work on Plasma Mobile is not visible yet because it is more than just a different UI. A whole new stack of software, which is leaner and more resource efficient, is needed for a mobile platform. According to Pintscher and Kügler, "We are spending a lot of time in optimizing memory usage to make Plasma run on these smaller systems. This work will benefit not only the future platform but also the existing software."

The community had a meeting in December 2017 where they gathered in a castle in Germany and spent a week optimizing Plasma. "It's boring work to the general public, but it's really important for people who have high expectations of their software."

The KDE team also plans to increase the focus on making KDE technologies more friendly to embedded systems. "We are partnering with a company that's building smart speakers, which is one of the fastest growing markets right now."

The KDE leaders admitted that the KDE community is not very good at marketing and getting the word out. "We are professionalizing our marketing department, but in the end, there is only so much we can do. In our opinion, the priority should be improving the software experience, because once you have a really good product, it will spread by itself."

Pineapple Fund

The Pineapple fund was started by an anonymous donor (with the nickname "Pine") who made a lot of bitcoin in the early days and is now donating to non-profit organizations he cares about. "KDE was one of the organizations that applied and was selected for the donation. We received $200,000. This is the biggest donation KDE e.V. has ever received. We will put it to good use," Pintscher said.

The donation has doubled the total funding that the community has to-date and they need to use this money carefully. One way to use that money could be by paying a developer to write code. But KDE is a volunteer-driven project; you can't pay one developer while the rest of the people are doing it for free.

"None of KDE's current money goes towards development. We do have one part-time paid position, but it's to assist the board and the community with organizational matters. There are two other contractors who work for our marketing team. Beyond that there are no people paid by KDE e.V. to work on the software."

Pintscher and Kügler said the community is discussing how to use this money in the most impactful way. "We are contemplating hiring or contracting people to do specific things, such as documentation, clean-ups, usability tests, or marketing research."

"What we are trying to do is invest in multipliers. We want to make it easier for volunteer developers to get involved. If we can pay one person to write code, it gets us only so far, but if we pay someone to make it easier for 100 developers to write code, that takes us much further. So we are looking into what will have the biggest long-term impact."

The community is also hoping to spend the grant on areas where they are not strongly supported by the current process. "Documentation happens to be one of the areas that is underrepresented, and it really eats the developers' time. It also needs a different skill set. If you ask a software developer to write user documentation, you sometimes get incomprehensible documentation, and you are also wasting the time of someone who would rather be doing something else. Plus you are leading the developer to quicker burnout. Working with some documentation professionals could help smooth out the whole development process, keeping it fun and more productive."

The KDE community greatly values this $200,000 donation and is looking very closely at how to maximize its impact. ▮▮▮

Author

Swapnil Bhartiya is a writer and journalist covering Linux and open source for more than 10 years. He is also a science fiction writer whose stories have been broadcast on Indian radio and published in leading Indian magazines. He founded an open source web magazine while living in Europe. Swapnil currently resides in Washington, DC.

Meet the open-source community's answers to Google Assistant and Alexa

Hey, Mycroft!

Voice-activated assistants like Mycroft bring online, hands-free help to users, but with more transparency and less spying. *By Claudio Cambra*

E very few years, Silicon Valley becomes obsessed with a new development in the tech space. Some, like the 3D web, fizzle out and leave nothing more than a footnote in the history of the tech industry. Others, like the push towards cloud services, have become a central part of many users' (and developers') lives.

Silicon Valley's latest craze is voice assistants. Google, Apple, Amazon, and a few others are all jostling to make the most capable voice assistant on the market, hoping to create an artificial intelligence that will do everything for you on your phone or computer. This year's CES has seen all kinds of devices powered by either Google's or Amazon's AIs – home automation kits, smart speakers, cars, and more.

As is the case with most proprietary services, these AIs have one big caveat: They are opaque, and they collect a lot of information about you that gets sent to someone else's opaque servers. Luckily, the open-source community is already at work on several voice assistants that are transparent and respectful of your privacy – and that strive to match their commercial rivals in functionality.

Mycroft

Mycroft [1] is the most developed and most popular open source AI assistant. It is created by Mycroft AI, a 19-man operation set up by entrepreneur Joshua Montgomery in Kansas [2]; despite its small size, Mycroft AI has the ambitious plan of creating a virtual assistant that will be able to compete with the devices developed by the giants of Silicon Valley (Figure 1). The company has announced plans to "demonstrate something approximate to a human" by 2020 [3]. The open source development model is an integral part of this plan.

The clearest sign of Mycroft's intent to take on Google and Amazon is their devices. The company has released one smart speaker, the Mark I, for developers. Funded via Kickstarter in 2015 [4], the device resembles an adorable cross between ET and Eve from the movie *WALL-E*, with a white ovoid shell and two big eyes and a mouth made with LEDs at the front of the device (Figure 2). The company has started another Kickstarter campaign for their Mark II, which resembles a plastic and cloth egg with a screen on the front [5].

Figure 1: Joshua Montgomery, Mycroft AI's founding father and CEO.
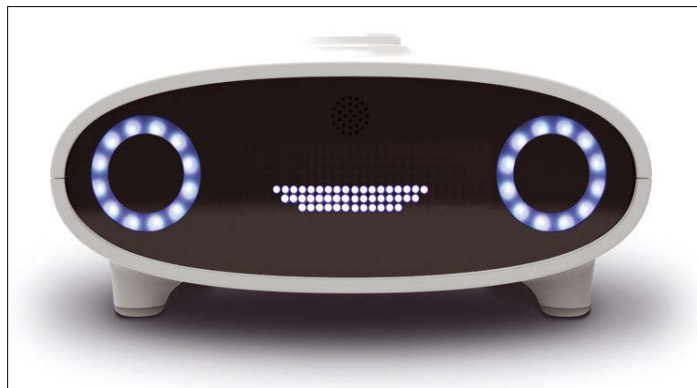
Figure 2: The Mycroft Mark I, with its big LED eyes and white plastic shell.

Despite their commitment to developing smart speakers, the company also provides its software for Linux distributions.

Skills

Mycroft organizes tasks into *skills*. When the user asks a question or gives a command, Mycroft uses an intent parser to identify the reason for the utterance. Once the intent is known, Mycroft checks to see whether it has a skill that addresses the intent. For instance, if you ask Mycroft about the weather, it will parse the request and send the intent data to the weather skill. Mycroft currently installs with 30 *core skills* when you set up the system [6]. You can think of these skills as the things Mycroft "knows how to do" at the time you install it. (See Table 1 for some examples of Mycroft core skills.)

```
git clone https://github.com/↵
MycroftAI/mycroft-core.git
cd mycroft-core
bash dev_setup.sh
```

Starting

Now that the installation is complete, all that is left to do is to run Mycroft. Start Mycroft with:

```
cd mycroft-core
./start-mycroft.sh debug
```

These simple commands will initialize all of Mycroft's core services (the microphone listener, the core skills, etc.) and will bring up an interface that you can use to troubleshoot the assistant and perform a limited set of basic commands (Figure 3). Setup, however, is not complete – and this fact reveals one of Mycroft's major caveats.

The Cloud

For Mycroft to actually function as a virtual assistant, it must be linked with Mycroft AI's online account service [9]. Opening an account is relatively painless, requiring a Google or Facebook account or a conventional email and password combination. Through the website, you can then pair the desktop client with the account service using a pairing code that you can acquire by asking "Hey Mycroft, pair my device." After this process is complete, Mycroft gains access to its core skills, such

The beauty of the open source development model is its flexibility and capacity for expansion. Unlike many proprietary digital assistants, Mycroft lets users develop their own skills. The project website includes instructions for creating your own Mycroft skill [7].

Users can submit these custom skills for acceptance and distribution to the Mycroft user community. The website has a long list of community-developed skills that any user can add to their Mycroft configuration. Examples of community-developed skills for Mycroft are shown in Table 2. Some of these skills are currently marked as untested or works in progress, but the examples in Table 2 give an indication of the kinds of tasks Mycroft users are hoping to automate.

This extensive list of core and community skills means you can tailor Mycroft to do what you need it to do. Of course, as you will learn later in this article, some of these skills don't exactly work as advertised at this point in Mycroft's evolution, but if the power of open source development leads to community bug hunting and a vibrant pipeline for newer and better community skill development, the quality and scope of Mycroft's capabilities will progress rapidly.

Installation

The main reason for Mycroft's desktop version is to make it easy for developers to develop for the Mycroft platform [8], but you don't have to be a developer to get it to work. It takes just a few commands to get Mycroft up and running.

Table 1: Mycroft Core Skill Examples

Skill	Description
Audio Record	Record and play audio
Date Time	Tell the date or time
Desktop Launcher	Open a desktop application (e.g., "Open Firefox")
DuckDuckGo	Query the DuckDuckGo search engine for general questions
Joke	Tell jokes ("Tell me a joke")
NPR	Listen to news from the American National Public Radio network
Reminder	Ask Mycroft to remind you of something ("Remind me to turn off the oven in 15 minutes")
Stock	Look up the stock price for a specified company
Weather	Report on the current weather and weather forecasts
Wiki	Look up an entry in Wikipedia

Table 2: Community Skill Examples

Skill	Description
bitcon-price	Checks the current price of bitcoin
calculator-skill	Solves mathematical problems
daily-meditation	Plays a daily meditation from the *Daily Meditation* podcast
facebook	Generates posts for Facebook
Google-calendar	Check and add Google Calendar events
Google-gmail	Get email from your Gmail inbox
home-assistant	Control your home devices using the Home Assistant open source home automation platform
kodi-cadair	Playback and search with Kodi media player
youtube	Control and listen to a YouTube video
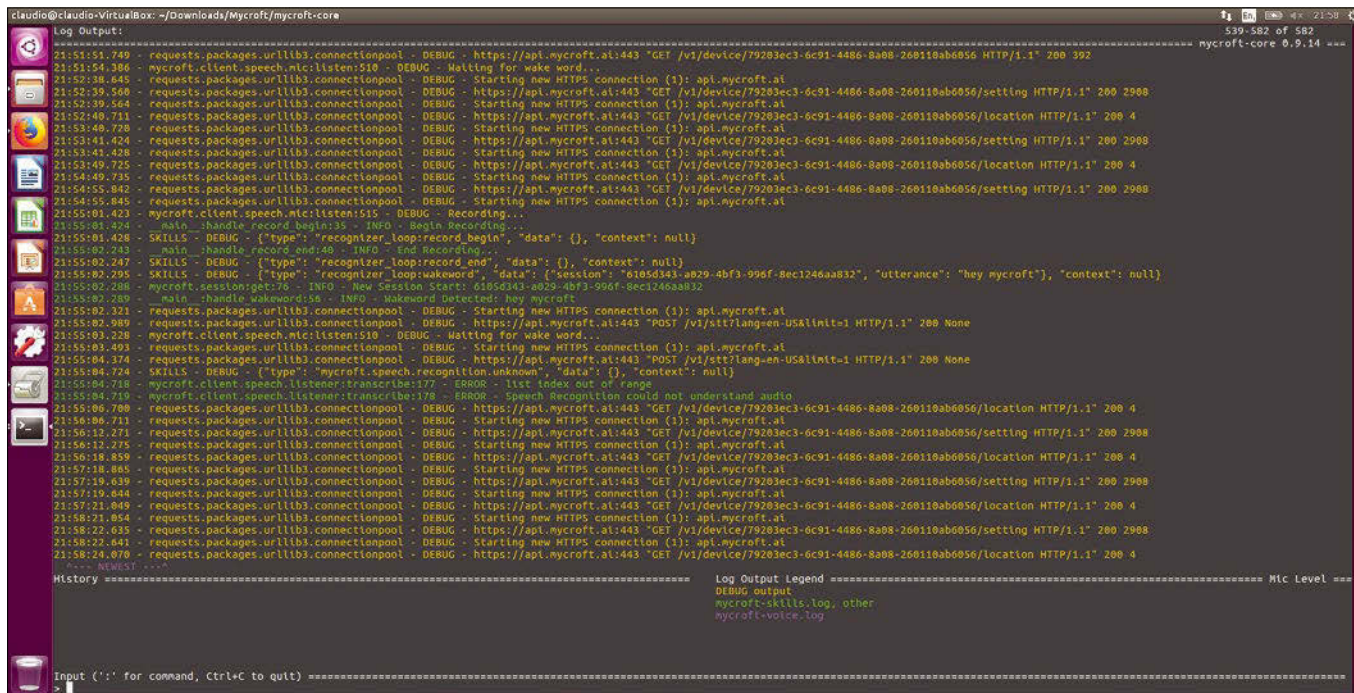
Figure 3: Mycroft's CLI interface for troubleshooting.

as reporting the weather and answering easy questions. The website also allows you to adjust some of Mycroft's settings, such as its voice, units of measurement, and more (Figure 4).

Mad Skillz

Mycroft's modularity is one of its strong points. Mycroft AI encourages developers to write new skills for its assistant [10],

and users are free to add or remove skills as they wish. Mycroft AI has created a convenient script that makes it easy to add a skill [11]. To download this tool is as simple as typing the following command in your terminal:

```
git clone https://github.com/MycroftAI/msm.git
```

To install a skill using the Mycroft Skills Manager (msm) script, all you have to do is:

```
cd msm
sudo -H ./msm install <skill name>
```

Just a tip – msm will require that you have the *python3-dev*, *python3-setuptools*, and *python3-pip* packages installed.

Privacy

Mycroft's reliance on open source software on almost all of its layers makes it more comforting to use than closed-source alternatives. Most of the components are open source: Its wake-word recognition is powered by PocketSphinx, for example, and its skills framework is coded in Python. Mycroft has even developed its own open code for language processing, Adapt, to avoid relying on proprietary software for its product.

However, Mycroft's by-default use of the online Mycroft Home service may concern those who are privacy-conscious. Some of the information sent home, such as language preference, may
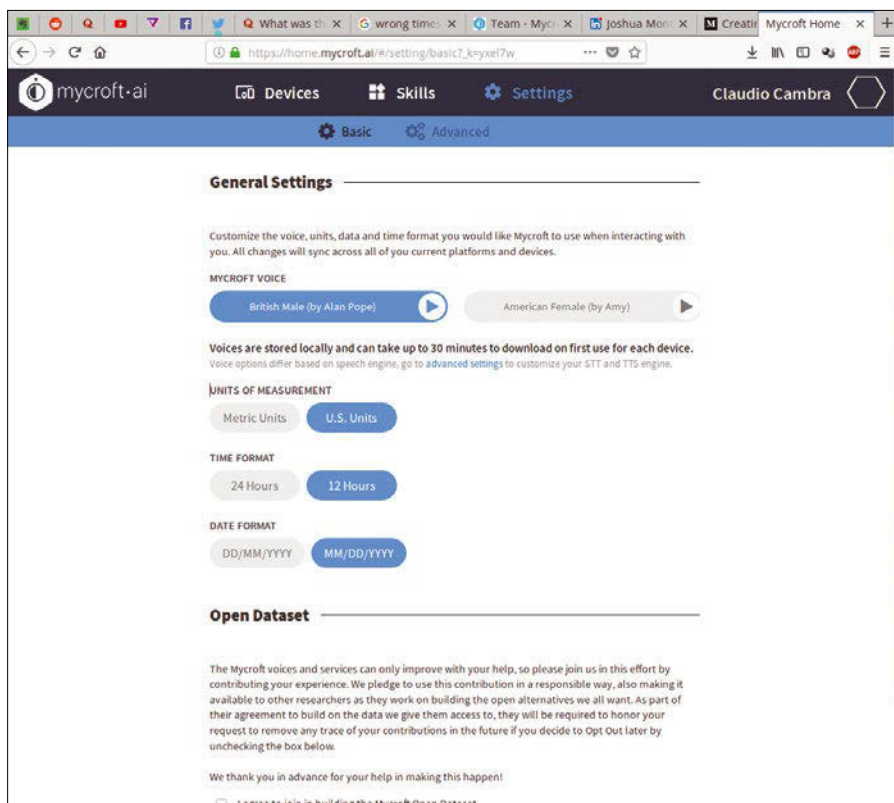
Figure 4: Mycroft Home's website allows you to change some of Mycroft's settings to tailor the system to your tastes.

seem relatively harmless; but other details, such as location data, could be too much for some users.

When it comes to it a Speech-to-Text (STT) platform, Mycroft prioritizes Google's STT platform but allows users to pick from a handful of other solutions (Figure 5). Google's STT system, a cloud-based engine owned by a company not known for its concerns over users' privacy, might make some users uncomfortable; Mycroft AI has therefore emphasized its decision to allow users to choose which engine they'd like to use. The company has been quick to assure users that the assistant will be given the option to also employ Mozilla's open-source DeepSpeech service by the end of March 2018 [12]. Mycroft AI insists that all the data it sends to third-party STT services is anonymized, with the service only seeing that Mycroft sent an audio file; but for some, this precaution might not be enough.

Work-in-Progress

It's important to say that, despite the fact that Mycroft is the most popular AI assistant in the open source field, it is by no means a finished piece of software. One shouldn't expect a service on the level of Alexa or Siri (yet).

The first issue you'll likely encounter is that Mycroft often fails to hear your wake-up call (Hey, Mycroft!) and, at least in my case, leaves you screaming at your computer like a deranged Sherlock Holmes. Sometimes it will hear you and work fine for the first few commands and then suddenly give up on you. It's very frustrating, but it's likely that as usage of the assistant increases, the developers will improve on this significantly.

When it did hear me, it often failed at delivering the answer I was looking for. Some fairly basic tasks, such as telling the weather, would leave it thinking for a bit before telling me it didn't know how to answer (despite the fact that the weather skill was installed). In some other cases, it would give me a definition of what the weather is! Other simple requests, such as asking Mycroft who the president of the United States is, would also give me a definition of the head of state rather than telling me who the actual person is. A quick search on YouTube will provide more examples of Mycroft's hilarious answers, which range from bizarre to downright wrong [13].

To be fair, though, voice recognition was consistently good when it did listen, easily understanding what I said. It recognized what I said and, more importantly, understood what to do afterwards… most of the time.

Though it might sound like a nitpick, Mycroft's voice could also use some improvement; to say that it sounds robotic would be a kindness. He (or she, depending on which voice you choose) sounds like a cross between a Cyberman from *Doctor Who* and a prison warden. Depending on what he is saying, the voice can be either slightly amusing or slightly intimidating.

Honorable Mentions

As is always the case with the open source community, Mycroft is far from the only project under development. A significant number of virtual assistants are currently in development, all with slightly different focuses and abilities.

For the more private individuals, Open Assistant [14] is an interesting option. It is entirely offline and works only with open source components. Though Open Assistant is not as well known as Mycroft, it is a compelling alternative, but it is rather spartan in its abilities. It has, however, quite a few skills that
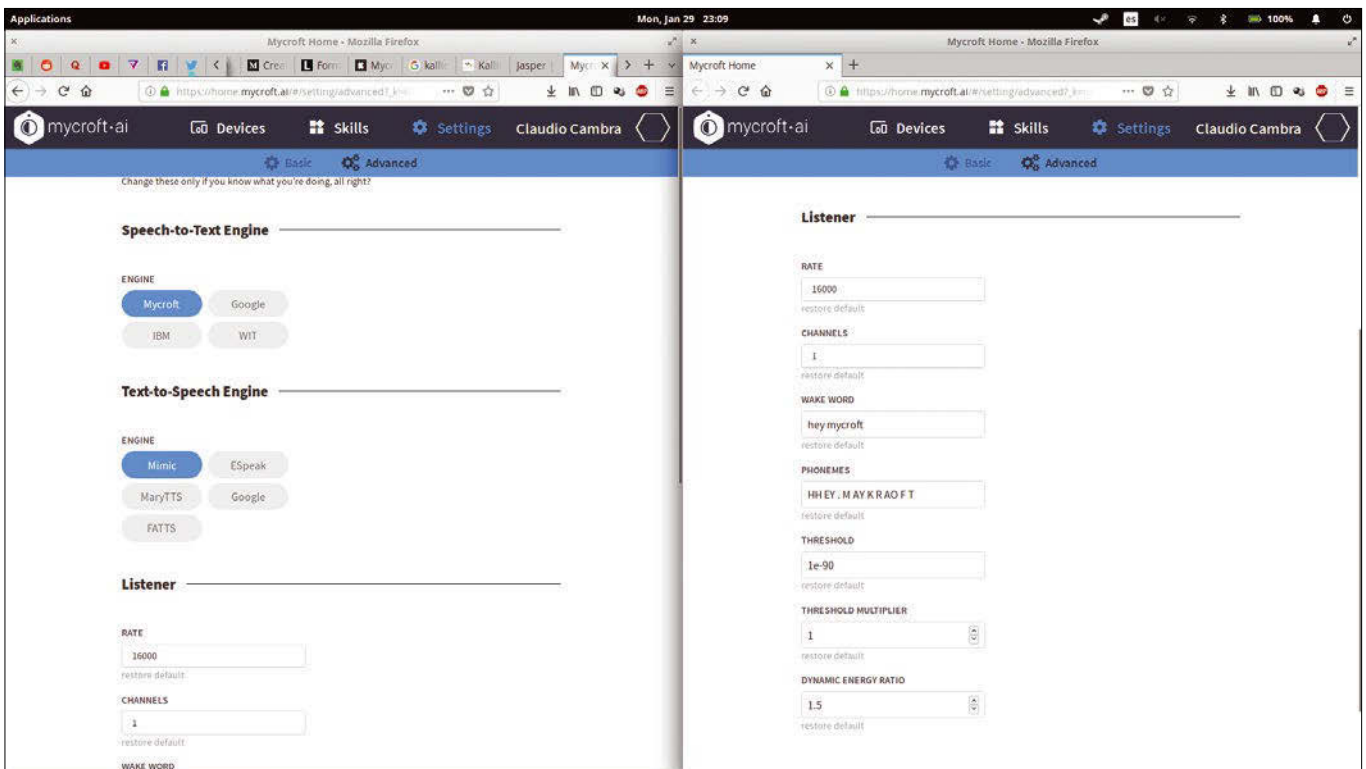

Figure 5: Mycroft Home also allows users to change a few advanced settings, including which STT service Mycroft uses.

could be useful for desktop users, such as the ability to open applications and resize windows; in that sense, it gets closer to being a genuinely useful computer assistant than many of its competitors.

Before creating their own personal assistant, the Mycroft developers had attempted to contribute to Jasper, which was one of the more well-known assistant projects in the open source scene. Jasper is meant to run on a Raspberry Pi and serve as an assistant, much like Mycroft's Mark I and II, rather than playing the role of a general-purpose desktop assistant on a PC; it still features very basic functionality, such as reading the weather and telling the time [15].

Another name that often pops up is Dragonfire [16]. The focus of Dragonfire, however, is much more particular than Mycroft: The project is intended to serve as a virtual assistant for motorcycle helmets rather than a general-use assistant for the home. Even so, the developers provide a package that will install on Ubuntu… but only Ubuntu. Any other distro will require a compilation. Dragonfire is notoriously difficult to get working – the `.deb` file provided by the developers failed in our attempted installations.

Kalliope is yet another project aiming to provide a functional and easy-to-use assistant, but this time with the focus on home automation [17]. Like Open Assistant, it can also be entirely self-hosted, though it also supports other cloud-based STT engines such as Google's. Development is very active, and, much like Mycroft, it also has skills (called *neurons* in Kalliope's case) that you can add to increase its functionality. Kalliope aims to support a number of Internet of Things (IoT) devices and, unlike most of the other assistants, it has a mobile app you can use to control your home.

In addition to these assistants that are officially billed as open source, some mainstream assistant tools, such as Alexa, Cortana, and Google Assistant, also contain some open source components (Table 3).

Conclusions

People like Elon Musk warn about the danger of unchecked development of artificial intelligence [18]. In the short term, these warnings might seem like paranoid delusion, but in the log run, leaving AI development in the hands of opaque governments and corporations is simply asking for trouble. As always, the smarter and safer bet is open source.

Leaving aside the more philosophical debates surrounding artificial intelligence, there is no question that Amazon,

Google, and Apple are much farther ahead with their virtual assistants than the open source community is. Mycroft is by no means a shoddy piece of software, but it is nowhere near the same level of polish and capability that comes with a tool like Alexa. But that's fine; it's not supposed to be. The future will see whether the Mycroft developers reach their lofty goals, but more importantly, it is essential that an open source alternative to the proprietary solutions continues to exist. There are plenty of walled gardens in the technological space already; ensuring that some open meadows flourish in the evermore important realm of artificial intelligence is of paramount importance to ensuring our privacy.

Just don't ask Mycroft what beans are. ∎∎∎

Info

[1] Mycroft homepage: *https://mycroft.ai/*

[2] Mycroft team: *https://mycroft.ai/team/*

[3] Mycroft in AI XPRIZE: *https://mycroft.ai/blog/mycroft-ai-x-prize/*

[4] Mycroft Mark I: *https://mycroft.ai/mark1/*

[5] Kickstarter for Mycroft Mark II: *https://www.kickstarter.com/projects/aiforeveryone/mycroft-mark-ii-the-open-voice-assistant*

[6] Mycroft skills: *https://mycroft.ai/documentation/skills/*

[7] Creating a Mycroft skill: *https://mycroft.ai/documentation/skills/developing-skills/*

[8] Mycroft's GitHub page: *https://github.com/MycroftAI/mycroft-core*

[9] Creating a Mycroft account: *https://home.mycroft.ai/#/signup?_k=n47t5k*

[10] Mycroft skills repository: *https://github.com/MycroftAI/mycroft-skills*

[11] Mycroft Skills Manager: *https://github.com/MycroftAI/msm*

[12] Mycroft on moving to Mozilla's DeepSpeech: *https://mycroft.ai/blog/mycroft-speech-to-text-and-balance/*

[13] Asking Mycroft what beans are: *https://youtu.be/D5J7vVQNkCw*

[14] Open Assistant: *http://openassistant.org/*

[15] Jasper: *https://jasperproject.github.io/*

[16] Dragonfire GitHub: *https://github.com/DragonComputer/Dragonfire*

[17] Kalliope: *https://kalliope-project.github.io/*

[18] Elon Musk on the dangers of AI: *http://www.independent.co.uk/life-style/gadgets-and-tech/news/elon-musk-artificial-intelligence-openai-neuralink-ai-warning-a8074821.html*

Table 3: Open-Source Components

Component	Mycroft	Alexa	Cortana	Google Assistant	Siri
Wake-word	PocketSphinx, moving to Precise	Alexa Voice Services	Microsoft Speech API	Google Cloud Speech API	N/A
STT	OpenSTT, Google STT, moving to Mozilla DeepSpeech (aiming for March 2018)	N/A	Microsoft SAPI, Project Oxford	Google STT	N/A
Natural language understanding	Adapt and Padatious	Alexa Voice Services	LUIS	N/A	N/A
Text-to-Speech (TTS)	Mimic (based on CMU Flite)	Polly	N/A	Google TTS	N/A
Skills framework	Python	Lambda framework, functional programming	Cortana Skills Framework, node.js, or .NET	DialogFlow	N/A

OPEN SOURCE MONITORING CONFERENCE

NOVEMBER 5 - 8, 2018 | NUREMBERG

NEW APPROACHES, PIONEERING SOLUTIONS AND FUTURE PERSPECTIVES

CALL FOR PAPER & EARLY BIRD

JUNE 30, 2018

www.osmc.de

A lightweight Linux with excellent cloud connectivity

Peppermint

If you are looking for a lean but modern Linux with a solid foundation and seamless cloud connection, take a look at Peppermint OS. *By Erik Bärwaldt*

Modern Linux users expect a robust system complete with office suites, cloud services, multimedia applications, and games. Gathering all these requirements under one roof requires lots of space for a large number of available applications.

Unfortunately, resource consumption increases with software requirements, so older or less powerful computer systems quickly reach their limits. Peppermint OS is an attempt to package the tools of a contemporary OS into a distro that will run on old hardware. The result is an appealing mix of lean and partly self-developed software based on Lubuntu.

Peppermint OS aims to make a good impression on legacy hardware. In addition to the 1.2GB image for state-of-art 64-bit systems, the developers also offer an ISO image for 32-bit computers [1]. Both variants of the system, which are based on the code of Ubuntu 16.04, will be supported until 2021.

The hybrid image of Peppermint OS is suitable for both DVDs and USB sticks. After creating and inserting a startup medium, you will see a GRUB 2 menu, which offers Live boot or installation on a storage device.

Live

We started out in Live mode and used a USB stick as the starting medium. The operating system launched quite slowly on our ancient test device; the GRUB menu remained on the screen for a long time, initially without revealing any computer activity.

The system boots to an LXDE desktop in a dark color scheme. A panel appears at the bottom of the screen, and the desktop has a single icon to let users launch the installation. Click *Menu* in the lower left corner of the panel to open an ergonomic Whisker menu (an equivalent to a Start menu commonly used on Xfce and LXDE desktops).

The menu includes numerous launchers for opening cloud connections or activating online services: In the Graphics submenu, you will find Peppermint OS apps from Pixlr [2]; the Internet subgroup contains a Dropbox client, and the Office menu has launchers for Gmail, Google Drive, and Google Calendar.

In addition to these launchers for online services, you'll find a collection of Linux desktop applications, with heavyweights such as LibreOffice, Firefox, Thunderbird, and Gimp missing. Such resource-hungry programs would unnecessarily bloat the operating system and undermine the goal of running smoothly on older hardware. The only large standard applications are the current VLC media player and the Chromium web browser.

The basic components of the system are fully up to date. Peppermint OS offers the 4.10 kernel, version 1.18.4 of the X server, version 2.23 of the GNU C libraries, and systemd 229.

Icy

The *Ice* option in the *Internet* menu is a special program for integrating Internet addresses into the menu structure. *Ice* lets you add a URL and an intuitive name for the web page to the application menu (Figure 1). If you click on the

Lead Image © pixelrobot, 123RF.com
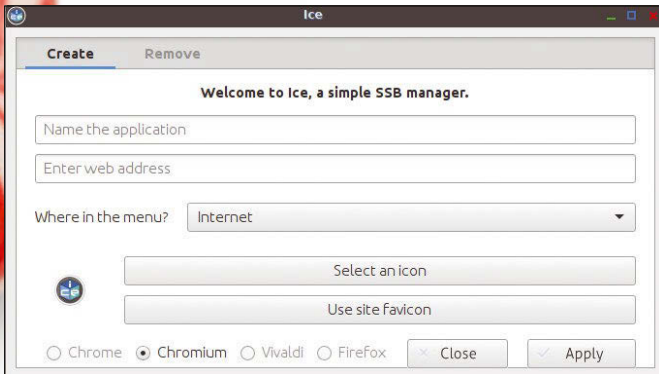
Figure 1: Ice integrates websites into the application menu.

Figure 2: The Settings panel combines all configuration tools.

menu item, the web browser automatically opens the web page.

Getting Started

If you choose to install Peppermint OS on your storage device, click the *Install Peppermint 8* icon on the LXDE desktop. Ubuntu's standard Ubiquity installer launches, and it sets up the operating system on the computer in a few steps.

The Settings menu lets you configure and customize your system. The Peppermint Settings Panel combines almost all important configuration tools in an intuitive interface. Some of these tools are courtesy of Xfce.

Another special feature in the settings dialogs is the Peppermint Control Center, which you can call in the *User* tab of the Settings Panel (Figure 2). The Control Center helps with configuring input devices, desktop effects, and the window manager.

A small application named Advert Blocker helps keep annoying online advertising off the system (Figure 3). Advert Blocker, which is from the antiX Linux inventory, has a graphical interface and supplements the /etc/hosts file in the system with a whole series of addresses for known display networks and tracker systems. If a URL is on the list, requests to the URL draw blanks.

Avert Blocker uses freely available and permanently updated lists of servers from the marketing industry. You can currently choose among four list providers and can even combine lists.

The program downloads and installs the server lists, which primarily cover marketing pages from English-speaking countries and offer only limited benefits for users in other countries. You might want to install an additional ad blocker such as uBlock Origin [3].

Software

The installed version of Peppermint OS offers almost the same software inventory as in Live mode. You have a choice of more than 55,000 packages from Ubuntu repositories – in addition to the countless binary package archives available through individual projects. Peppermint OS comes with two graphical package management tools: In addition to Synaptic, you will also fi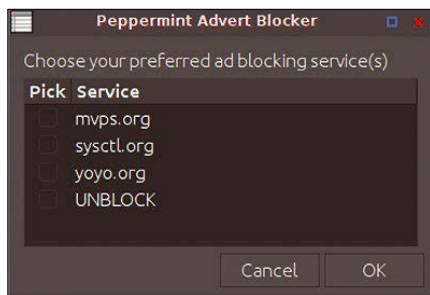nd the Software Manager tool with individual application groups from Linux Mint (Figure 4). Synaptic is aimed more at advanced users,

Figure 3: Keep out: the Peppermint OS ad blocker.

and Software Manager provides beginners with a better overview of the available applications.

You can reach Synaptic through the System Tools menu. Call the Software Manager through the Settings Panel.

In Theory

Peppermint OS views itself as a system for legacy computers, so we decided we'd better put it through its paces using old hardware. We used two 10-year-old laptops with 32-bit CPUs, one with a single-core processor and one with a first-generation Centrino dual-core processor. We tested the 64-bit Peppermint OS version on a device with a Core 2 Dual CPU.

On the two 32-bit computers, the long loading times for the system were noticeable in Live operation; this effect was not due to the processors but to the very slow hard disks (by today's standards). The hard disk proved to be the bottleneck even after the permanent installation. With such old hardware, it is therefore recommended to use the fastest possible mass storage to ensure agile system behavior. The main memory utilization, on the other hand, was quite low at 350 to 500MB each.

Even on the oldest test device, an HP Compaq nc6220 with an integrated GMA900 series Intel graphics card, which is poorly supported under Linux,

Peppermint OS succeeded in conjuring up optical effects such as transparencies on the screen without pushing the graphics card to its performance limits. Movies played in full screen mode with VLC generated a high system load but ran smoothly. Only video transcoding, which requires instruction set extensions such as SSE4.1, overtaxed the old computer systems. Image processing tasks sometimes took the hardware to its performance limits.

Conclusions

Peppermint OS is a modern and innovative Lubuntu derivative that combines a full desktop with mobile technologies and some interesting innovations. In our tests, the system fulfilled its claim of making optimum use of legacy hardware. If there is anything to criticize, it would be the sometimes-inconsistent integration of applications into different menu groups. If you want to continue using an older computer and are looking for a powerful all-round system with seamless online integration, consider Peppermint OS. ∎

Info

[1] Peppermint OS: *https://peppermintos. com/guide/downloading/*

[2] Pixlr: *https://pixlr.com*
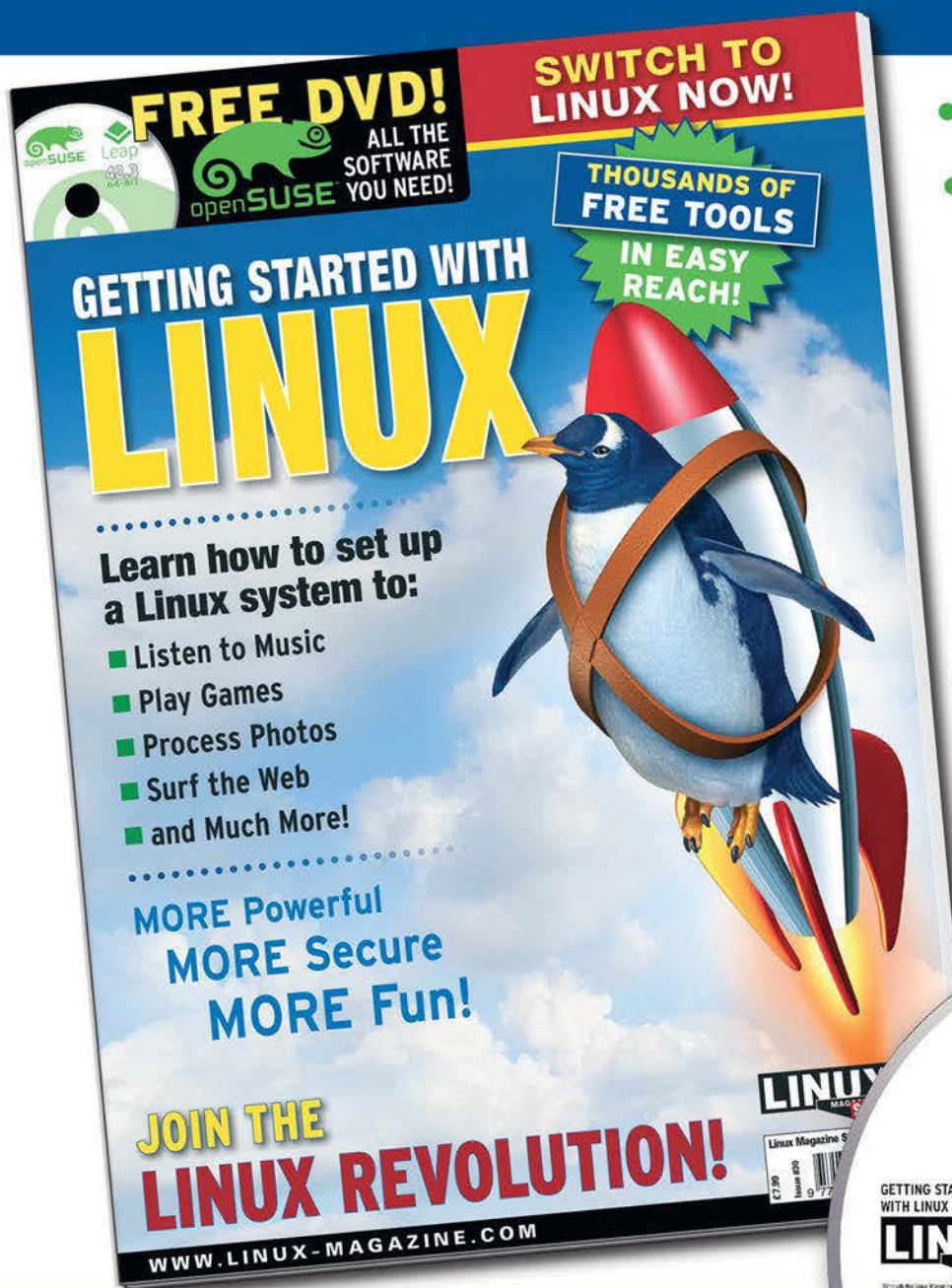
[3] uBlock Origin: *https://github.com/gorhill/uBlock*

Figure 4: Suitable for experienced users and beginners: Synaptic (bottom left) and the Software Manager (top right).

Testing scanners under Linux

Scan It

Using scanners to digitize documents in Linux was once a perilous business, but Linux scanner technology has improved in recent years. We study the state of the art with some leading scanners from various manufacturers. *By Erik Bärwaldt*

I f the paperless office were already a reality, no one would need a scanner anymore. But wherever you look in the corporate environment today, you very quickly see that traditionalists continue to defend the ubiquitous role of scanners in the office. And

Big Devices

Many manufacturers offer large stand-alone devices with scanning units, some of which also work with Linux. However, under Linux these multifunction devices are mostly used as pure printers for workstations on the intranet. They are not suitable as scanners for ordinary client systems.

Since such business devices are primarily aimed at large departments or branches, the scanning units are mostly used to digitize documents directly on the device and then integrate them into the company's internal workflow. This is usually done via touchscreens, often with an integrated keyboard and corresponding menu navigation. The digitized documents can then be mailed or integrated directly into a groupware solution.

not only that, increasing numbers of multifunction printers with integrated scanning units are being used in home offices and by private users.

Electronics stores continue to sell dedicated flatbed scanners, as well as equipment for special applications. This includes devices for mobile use, but also document scanners that automatically digitize large volumes of documents (see the "Big Devices" box). This article investigates how this device farm gets along with Linux.

Standards

When scanners gradually became affordable for smaller companies in the early 1990s, a consortium of various hardware and software manufacturers developed the TWAIN specification. It defines a uniform interface for controlling scanners from within applications. While the TWAIN interface was initially only developed for Microsoft and Apple operating systems, version 2.0 also has supported Linux [1] from 2008.

At the same time, Linux developers have been working since 1996 on a

unified interface for imaging devices called Scanner Access Now Easy (SANE) [2], which they released under the GPLv2. It is divided into a front end and a back end, with the two being strictly isolated. The back end forms the interface between the operating system and the scanner or camera hardware, and the front ends integrate into corresponding applications, making it possible to control the hardware and use its capabilities.

Thanks to this concept, all common front ends cooperate with the SANE back end without time-consuming adjustments. This explains why SANE is now considered the quasi-standard for scanners under Linux. Graphical applications like XSane [3], xscanimage [4], Simple Scan [5], or KDE Skanlite [6] all rely on it.

VueScan [7] is a commercial solution for Linux, which American Ed Hamrickhas been working on since 1999. It uses proprietary drivers and drivers from scanner manufacturers. The software supports more than 4,500 scanner models under Linux and is avail-

Lead image © wbraga, 123RF.com

Gobbledygook: CIS or CCD?

In the technospeak of every type of scanner's technical specifications, prospective customers will repeatedly stumble over two abbreviations: CIS and CCD. Both terms denote different constructions of the scanner's optical reading units.

Charge Coupled Device (CCD) technology uses a cold light tube as the light source, which directs the light onto a prism and then splits it into the primary colors red, green, and blue. The CCD sensor then captures each color with its own sensor line and converts it with light-sensitive capacitors and converters into numerical values that can be read by computers.

In contrast, Compact Image Sensor (CIS) scanning units rely on red, green, and blue light LEDs as light sources that use a simple rod lens to direct the light to the sensor, which in turn converts the light into computer-readable signals. Due to its simple design, the CIS scanning unit is significantly smaller than a CCD unit and also requires significantly less operating energy, but it does not provide any depth of field due to its simple design. CIS scanners cannot therefore digitize thick documents, for example books, because they do not lie flat on the scanner's glass pane.

able in two versions. Both offer the same GUI, but differ in terms of functionality and price [8].

In addition to the two quasi-standards, manufacturers of scanners and multifunctional devices offer their own front ends, which generally use SANE as the back end. Some of them provide their own modules for SANE in order to make hardware that has only been on the market for a short time usable under Linux.

Test Subjects

In our lab, *Linux (Pro) Magazine* looked at devices from a wide variety of applications. The field extends from multifunctional devices with scanning units for small and medium volumes through dedicated scanners for use in small offices, to two extremely compact models for mobile use. In addition to SANE and VueScan, the test – if available – also takes into account software solutions from the manufacturers. The test field does not include special-purpose document and mobile scanners or barcode scanners. Devices that only digitize photos and are intended for private users are also missing.

Brother DS-820W and DS-920DW

The mobile hand scanner DS-820W [9] by Japanese manufacturer Brother costs around EUR185 (~ $160) and is designed as an all-round feed scanner with CIS sensor technology for mobile users (see the "Gobbledygook: CIS or CCD?" box). Thanks to an exchangeable lithium-ion battery, it even works without a stationary energy source. With the

aid of a motor-driven feeder, the device can read individual pages up to A4 format, but it can also digitize plastic cards, such as ID cards, driver's licences, or customer cards.

The model is similarly flexible when it comes to storing and transferring data. The scanner talks to the host computer either via WLAN or a USB connection. There is also a slot for an SD memory card to store the digitized data. This means that the scanner can be used as a standalone device.

Brother offers Debian and RPM packages with Linux modules for download [10] for the device, which weighs around one pound. The modules are available for both 32- and 64-bit architectures.

The almost identical model Brother DS-920DW [11] costs around EUR190 (~ $180), only a few Euros/dollars less than the DS-820W. It differs from its little brother only in terms of a slightly larger housing with a slightly different arrangement of the control elements and the possibility of scanning both sides of a page in a single pass (Figure 1).

The small device has a second CIS scan array, which you can switch on and off via the scanner's keypad. This explains why the DS-920DW weighs slightly more than the simplex scanner.

The driver modules are identical for both scanners. Larger infrastructures, in which both scanners work, do not require a separate module installation for each model. The installation of the driver module is very easy. First, download an appropriate deb or RPM package. Then connect the scanner to the host computer via a USB cable, switch to the command line, and install the SANE driver with admin rights as follows:

```
rpm -ivh --nodeps <packagename>
```

for RPM-based and Debian-based distributions, use the following command:

```
dpkg -i --force-all <packagename>
```

You then check to see if this step has integrated the module into the system by entering `rpm -qa | grep -e dsseries` or `dpkg -l | grep Brother`.

However, in the practical test, serious inconsistencies were revealed at times. We were able to install the modules on all test distributions without any problems, but some of them showed insurmountable weaknesses. The Russian ROSA Linux R8.1 failed to detect either scanner, although the module seemed to be correctly integrated into the system. The use of the two mobile scanners via the USB interface therefore failed under the ROSA Linux. The conversion of the deb packages to RPM format using the `alien` program was also unsuccessful.

Figure 1: Externally almost identical: Brother DS-820W (back) and DS-920DW (front).

Things looked different on Ubuntu 16.04 LTS and the Debian-based Q4OS (Scorpion) v2.4: The Brother scanners worked right away here. They not only achieved good scanning results with very different front ends, but also made a positive impression due to their very fast working speed.

Because it turned out to be difficult to integrate Brother's mobile scanners under ROSA Linux, I started additional tests with further distributions of both package management formats. These included Linux Mint 18.1, Mageia 6, and openSUSE Leap 42.2 and 42.3. The latest variant of ROSA Linux Desktop Fresh R10 was added.

The additional tests showed that the Brother models worked fine with all distributions that use the deb package format, while the driver installation via the RPM package format did not work on any distribution.

With massive overhead, it was finally possible to operate the scanners under openSUSE Leap 42.3. For this, you need to install the `/etc/udev/rules.d/55-libsane.rules` file with a text editor and add the following two lines at the end

```
# Brother
ATTR{idVendor}=="04f9", MODE="0644", ⏎
  GROUP="lp", ENV{libsane_matched}="yes"
```

After rebooting, the system recognized the scanners without root privileges, and XSane and Skanlite were able to use the devices normally.

After installing the driver, the `scanimage -L` command executed with root privileges reveals whether other RPM-based distributions also detect the Brother scanners. Brother's hopelessly outdated support page, which pays homage to long lost distributions like Mandriva 2010 or Red Hat [12], is a point of criticism. Manual adjustment of the Udev rules is also considered outdated.

Brother's support for Debian-based distributions also gives cause for criticism. Although the drivers provided worked without problems, the documentation is misleading. The support pages suggest various settings that are no longer necessary for current variants. Clearing out and updating the page would be helpful. Most of the information relates to distributions that are almost 10 years old and no longer in use anywhere. At the same time, more current Linux systems that came onto the market since 2012 are missing.

HP Color LaserJet Pro MFP M477fdw

Hewlett-Packard (HP) is one of the world's leading suppliers of dedicated scanners and multifunctional devices with a scan unit. In fact, the company covers the entire range of applications from private users to large corporations. In addition to a wide range of other operating systems, HP also supports Linux as a platform.

The constantly updated HP Linux Imaging and Printing (HPLIP) [13] application provides a graphical front end, based on SANE and the CUPS print server, to make multifunctional devices usable under a uniform interface. HPLIP has been in the software repositories of virtually all leading Linux distributions for years and is constantly growing to include driver modules for the latest HP models.

We used a current multifunctional device from the Color LaserJet Pro MFP M477fdw [14] series (Figure 2), which costs around EUR400 (~ $430), for our test. It is a powerful multifunctional device with printing, scanning, and fax capabilities, in addition to a copy function. The model is suitable for a recommended print volume of about 4,000 pages per month (i.e., for smaller departments, branch offices, and small companies). As is usual with HP, the system requirements list compatibility with countless Linux distributions.

In order to use the device (which has only been available for a few months) with HPLIP, you need software version 3.15.11 or newer.

Only this version has the required SANE back end modules. Especially with older Linux derivatives used primarily as server operating systems, such as Debian 8 or Centos 6, it makes sense to install the offered updates, since the repositories still contain older versions of HPLIP.

Once the driver is installed, HPLIP usually automatically finds a connected and supported device at startup. If this is not the case, a window pops up to support manual configuration.

Since many HP printers and multifunction devices now have multiple connection options, you can set up devices that talk to the host computer via the USB port, an Ethernet port, a parallel port, or WLAN. To do this, start the configuration routine in the graphical user interface via the *Device | Setup Device*.

HPLIP identifies the connected device, and you can set it up in a few steps. The configured new arrival then appears in a narrow vertical column on the application window's left side. On the right, you can trigger various actions and initiate status queries in five tabs.

Ideally, the installation routine should install the CUPS printer driver for the printer and copy unit, the associated fax driver (if necessary), and the
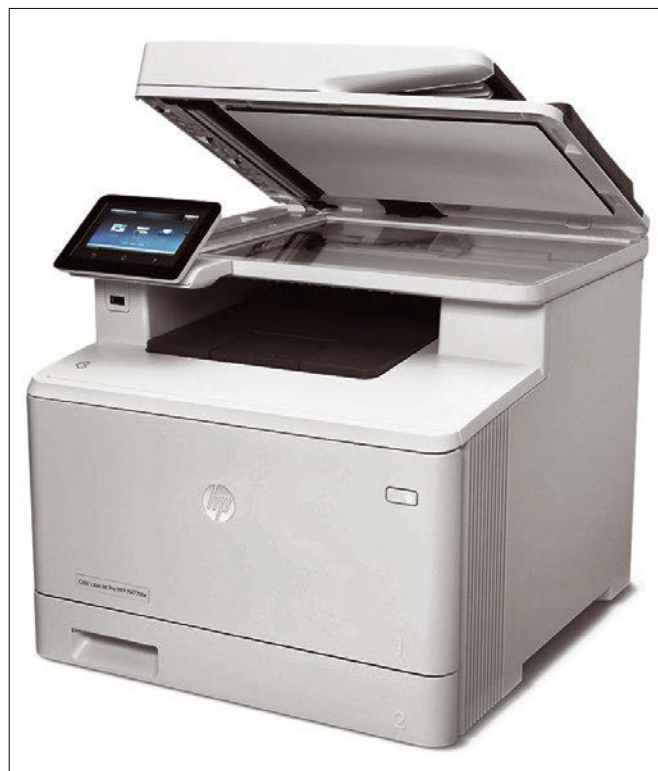
Figure 2: HP's Color LaserJet Pro MFP M477fdw also supports VueScan.
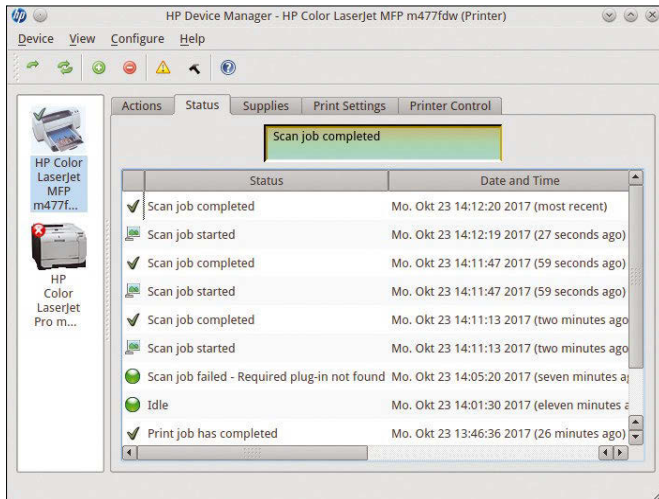
Figure 3: HPLIP also offers a status query to keep track of larger print jobs from the workstation.

appropriate scanner module in the background. Therefore, the first *Actions* tab also contains entries for printing and scanning and for manual control of the configuration.

Although the installation of the MFP M477fdw succeeded without obstacles, the first time an original was scanned, the scanner could not be initialized. The software required a specific scanner plugin that we were able to downloaded with a single mouse click. Afterwards, the scanning unit did its work without further complications; the drivers fully supported special functions such as automated scanning of several pages from the document feeder and duplex scans. In the *Status* tab, HPLIP also displays the device status. If errors occur while importing templates, they also appear here.

If you digitize large batches of documents via the single sheet feeder, it can help to check the status remotely. This way you can find out whether the scanner has processed a batch without having to physically check the scanner. However, since the status window also outputs messages from the printer, the display becomes somewhat cluttered when used intensively by several users (Figure 3).

The MFP M477fdw is also one of the few HP devices that supports the VueScan scanning software in addition to HPLIP. Its GUI offers far more detailed dialogs for setting up connected printers and scanners. In addition to the HPLIP functions, it allows image modifications such as rotation of the scanned docu-

ment or filter and color adjustments. The use of commercial software makes sense for demanding tasks (Figure 4).

In our lab, the MFP M477fdw also impressed with a sharp and detailed reproduction of the originals due to its high optical resolution of 1200 dpi. However, the CIS scanner digitizes books placed on the original glass with insufficient depth of field, and the document feeder is not suitable for bulky documents such as thick books.

HP ScanJet G3110

We tested one of the HP models without HPLIP support, the ScanJet G3110 flatbed scanner. It costs around EUR75 (~ $93), but it was already an end-of-life product when this issue went to press. SANE modules are missing for this scanner with a USB connection, but VueScan supports it as well as other standalone scanners.

The software uses the scanner's TWAIN capability. Manual setup is not supported. It automatically detects the scanner after it is switched on. If you uses several scanners in parallel with VueScan, you need to select the correct device in the program window's *Source* selection field.

Since the ScanJet G3110 also has a transparency unit for digitizing negatives or slides that can be controlled on other operating systems using push buttons located on the housing, the main interest
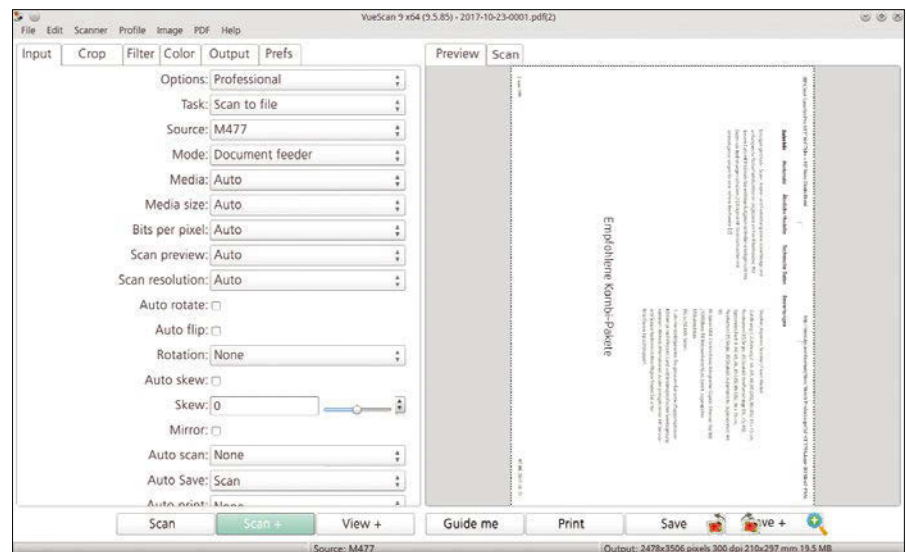
Figure 4: Because VueScan takes full advantage of the MFP M477fdw's feature set, the use of commercial software can make sense in some cases.

Figure 5: The ScanJet G3110 is a conventional flatbed scanner with a transparency unit.

here is how VueScan makes these special functions available (Figure 5).

We were able to use the ScanJet G3110 with VueScan without any problems, although some minor issues were revealed (Figure 6). The device works at a very leisurely pace when scanning and the device cover's function keys do not work with Linux. On the upside, the scanning process itself worked perfectly. The scanned content is detailed, sharp, and true to color, regardless of the storage format, even with very bad originals.

VueScan also correctly controls the transmitted light attachment without complaint: In the *Source* tab, you must change the *Flatbed* default setting to *Transparency* in the *Mode* selection field. This switches off the scan bar's light source and activates the transmitted light attachment located in the device's housing cover.

The light source's time-controlled automatic off switch also stands out: If the scanner does not read any documents for a long time, the software switches off the currently active light source so that the device does not unnecessarily waste energy. Since it also reliably wakes up from standby mode, the ScanJet G3110 delivers very good Linux support.

Samsung MultiXpress X4300LX and K4350LX

The South Korean conglomerate Samsung has gained a good reputation for multifunctional devices in recent years. The range of models offered extends from small systems for home networks or small offices to large floor-standing devices accessed by several hundred employees (see the "Samsung and HP" box).

Samsung has been delivering working Linux driver modules for most devices across all performance classes for years. For workstations that want to integrate
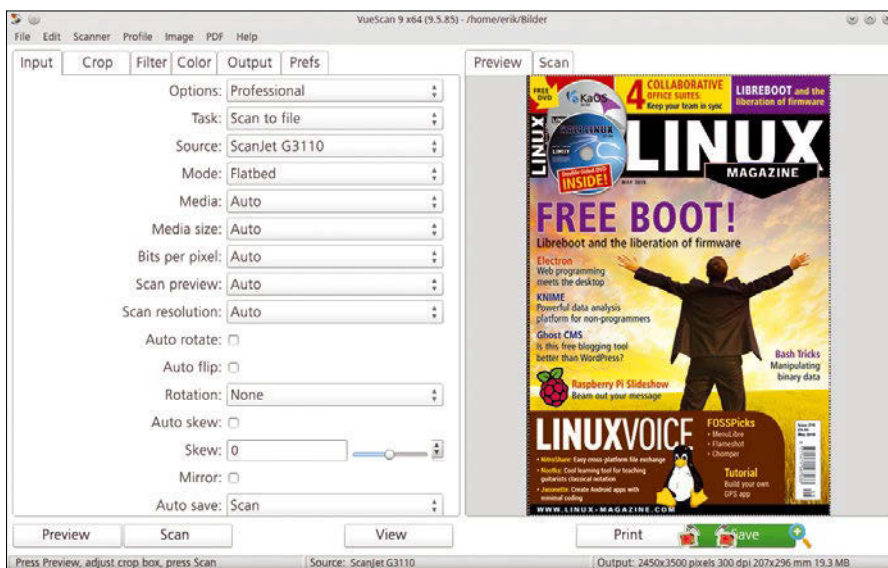
Figure 6: The ScanJet G3110 can also handle thick originals.

smaller systems, Samsung also offered a graphical front end.

For the scanner test, *Linux (Pro) Magazine* examined two brand-new large multifunctional devices from Samsung for workgroups or branches with several hundred employees: In addition to a MultiXpress X4300LX system for around EUR3,300 (~ $4,700) [15], we used an almost identical MultiXpress K4350LX [16] for compatibility testing (around EUR4,200 of $5,180).

The devices differ primarily in terms of the different printer units: The X4300LX has a color laser printer on board (Figure 7), while the K4350LX is a monochrome laser printer designed specifically for government agencies and public offices. Otherwise both models deliver quite impressive performance data: The scan units with duplex scanning arrays read up to 120 pages per

minute at a maximum native resolution of 600x600 dpi. A duplex document feeder is standard.

Numerous accessory options underline the devices' professional standard. They include an automatic sorter and stapler in addition to a base cabinet. One unit for automated accounting of copies supports credit card payment and additional paper feeds can be added.

Figure 7: The Samsung MultiXpress X4300LX is designed for large infrastructures.

The central unit itself consists of a single controller with a dual-core CPU with 1GHz clock frequency that is supported by 2GB RAM and a 320GB hard disk for documents and data.

However, the Samsung devices' firmware is out of the ordinary: Samsung relies on open standards and uses a customized Android version that meets professional users' security needs. This means that the multifunction devices can be operated like a conventional tablet PC via the built-in 10.1-inch touch display. You can even install additional apps for added flexibility.

Users with existing Internet access can send scanned documents to anywhere in the world using a one-time password and AES encryption. The documents can be sent to another computer, a smartphone, or a tablet PC via a peer-to-peer connection, regardless of the device.

Conversely, external devices can send documents to the multifunction device. For this purpose, Samsung has developed its own programs and free add-ons for all common web browsers, including Firefox and Chromium.

In this performance class, the two test devices were also clearly out of the ordinary in terms of Linux support: Samsung provides a Unified Linux Driver for the newer models in the form of two Deb packages (for 32- and 64-bit architectures) and a `tar.gz` archive [17]. The 15MB package contains modules for printers and scan units.

You integrate the modules into CUPS. To do this, execute the `./install.sh` setup script with admin rights. SANE then automatically detects the scan units, allowing them to be controlled by any front end.

The driver modules showed no weaknesses in the test: We were able to perform individual scans and to use the automatic sheet feeder and duplex unit. Especially when scanning large documents with many pages, the single sheet feeder showed its potential. The software on Ubuntu 16.04 LTS with Simple Scan couldn't keep up with the pace of the sheet feeder.

The scan results proved to be true to color and with good detail even without prior calibration of the scanner. The CCDM-MMT sensor technology developed by Samsung with extended color reproduction and excellent depth of field also makes it possible to digitize even poor quality originals without any problems.

VueScan mainly targets many smaller Samsung models but also some business devices. The MultiXpress X4300LX is one of them, but this does not apply to the USB interface. Therefore, the professional devices cannot be directly connected to single workstation systems with VueScan. If you want to use the multifunction device with VueScan, you have to integrate it into a network [18].

Conclusions

The availability of Linux drivers for scanners and multifunction devices under Linux is far better today than it used to be. SANE has established itself as a back end for controlling the devices. For this reason, most front ends, but also drivers developed by the manufacturer, integrate into the Linux ecosystem.

Nevertheless, some stumbling blocks still await you. Before purchasing a multifunction device or scanner, it makes sense to put large devices in particular through a functional test and carefully read the technical specifications. These devices often support Linux, but only in certain application scenarios. Scanners may only work in groupware solutions under Linux, or they may only provide support for certain interfaces such as Ethernet.

It is still the case that brand new appliances on the market generally still need a few weeks to mature. Only then can it be assumed that the available Linux drivers cover all functions required in everyday life.

However, it is gratifying that all manufacturers contacted by *Linux (Pro) Magazine* have adapted their Linux support to reflect the real world. When we pointed out weaknesses in their drivers, the manufacturers assured us that they would remedy the weaknesses quickly. ▮▮▮

Info

[1] TWAIN 2.0 specification: *http://www.twain.org/specification/archive/twain-2-0/*

[2] SANE: *http://www.sane-project.org*

[3] XSane: *http://xsane.org*

[4] xscanimage: *http://manpages.ubuntu.com/manpages/yakkety/man1/xscanimage.1.html*

[5] Simple Scan: *https://launchpad.net/simple-scan*

[6] Skanlite: *https://www.kde.org/applications/graphics/skanlite/*

[7] VueScan: *https://www.hamrick.com*

[8] VueScan prices and versions: *https://www.hamrick.com/de/reg.html*

[9] Brother DS-820W: *http://www.brother-usa.com/scanners/modeldetail/24/ds820w/overview?gclid=Cj0KCQjwh7zWBRCiARIsAId-9b4oZXukvELS-7FliZmSSTdbboQXRtoEH15eKEI7QPZ7ZaZ9wKO_8lgAaAvejEALw_wcB&ef_id=C9NO8rJvY3MAAE5x:20180412205633:s*

[10] Brother driver: *http://support.brother.com/g/b/downloadtop.aspx?c=us&lang=en&prod=ds820w_all*

[11] Brother DS-920DW: *http://www.brother-usa.com/scanners/modeldetail/24/ds920dw/overview?gclid=Cj0KCQjwh7zWBRCiARIsAId-9b4rWcSXpPBamlS_iSi7yTVKrhaw85B-gc4C1oWitCnHk3-KCzflwhlFl-aAo76EALw_wcB&ef_id=C9NO8rJvY3MAAE5x:20180412205834:s*

[12] Brother support page: *http://support.brother.com/g/s/id/linux/en/instruction_scn1c.html?c=us&lang=en&prod=ds920dw_us&redirect=on*

[13] HPLIP: *https://developers.hp.com/hp-linux-imaging-and-printing*

[14] HP's product page: *https://store.hp.com/us/en/pdp/hp-color-laserjet-pro-mfp-m477fdw*

[15] Samsung MultiXpress X4300LX: *https://www.samsung.com/us/business/KCDA_Cooperative_Contract/BLI%20Reports/Samsung%20MultiXpress%20X4300LX%20BLI%20Award.pdf*

[16] Samsung MultiXpress K4350LX: *https://www.samsung.com/us/business/KCDA_Cooperative_Contract/BLI%20Reports/Samsung%20MultiXpress%20K4350LX%20BLI%20Award.pdf*

[17] Samsung driver: *https://support.hp.com/us-en/drivers*

[18] VueScan and Samsung: *https://www.hamrick.com/vuescan/samsung_sl_x4300.html#technical-information*

Shared birthdays among party guests

Probability Party

At a party with 23 guests, having two guests with the same birthday in more than 50 percent of cases may sound fairly unlikely to amateur mathematicians. Armed with statistical methods, party animal Mike Schilli sets out to prove this claim. *By Mike Schilli*

The problem depends on the exact wording. Nobody can expect to go to a party with 23 people and meet someone with the same date of birth with 50 percent probability.

The unexpected result comes about by the fact that n guests are compared with each other (i.e., each with $(n - 1)$ other guests). It is much more likely that two random guests will be born in the same month and on the same day (the year is not considered) than if you only compare your own birthday with that of $(n - 1)$ guests [1].

Bottom Up

At a party with only two guests, what is the probability of both celebrating their birthday on the same day? Assuming a year to be 365 days for the sake of ease, without taking into account seasonal birth fluctuations or special cases such as twin parties, this occurs in one in 365 cases. Conversely, the probability that both guests have birthdays on different days is 364 in 365. If another guest joins the pair, the probability that no one in the room is celebrating their birthday on the same day is the coincidence of two independent events: The first event, which we just calculated to occur with the probability 364/365, and a second event, where the added person does not share a birthday with the first or the second person and can thus celebrate a birthday on only 363 of 365 days.

A statistician determines the total probability of different birthdays for three guests by multiplying the probabilities of the two independent single events above, which comes to 364/365 x 363/365, or around 0.991795. The reverse event, namely the case where two or more people celebrate their birthdays on the same day, results in a probability of 1 – 0.991795, or 0.008205.

The sequence continues with guests number four, five, and so on. In each round, the number of remaining days, and thus the numerator of the fraction, is decremented by one; the result of which is multiplied by the probability of the previous round.

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 go to *mschilli@perlmeister.com* he will gladly answer any questions.

Listing 1: birthday-paradox

```
01 #!/usr/bin/env python3
02
03 dates      = 365
04 dates_left = dates
05 prob       = 1
06
07 for person in range(1,24):
08   prob=prob*dates_left/dates
09   print("%2d: %.4f" % (person, 1-prob))
10   dates_left -= 1
```

Listing 1 [2] shows a lean Python implementation of the calculation. The output (Figure 1) shows that the 50 percent probability of a shared birthday between two guests was exceeded for the 23rd guest, showing a value of 50.73 percent. The script sets the number of days remaining in the calendar to 365 at the beginning and subtracts a value of 1 from it after each round, when a new guest with an unseen birthday arrives. The probability `prob` indicates the likelihood of no one in the room sharing their birthday with another person: For a room with only one person, this is obviously 1; for two, it is 0.9973.

In each iteration of the loop, Listing 1 multiplies the probability of the last round in `prob` by the new event's probability value and assigns the result back to `prob`. However, the probability of having no shared birthdays is not what we are looking for; instead, we want the opposite – the chance of a collision. Therefore, line 9 indicates the probability of the opposite event, or the chance of one or more people at the party sharing birthdays, or `1-prob`.

Simulator

A simulation script (Listing 2) will show whether the formula for the computation was correct; in each round it assigns 23 guests in the `guest_bdays` list to a party, assigns each of them a random birthday from a list of 365 integer values, and then decides in the `bday_match()` function whether there are integer duplicates in `guest_bdays`. The `randint()` function from the `random` module outputs values between the extremes 1 and 365 (inclusive) for birthdays.

The `for` loop starting in line 13 iterates over a total of 10 test runs with 100,000 parties each. For each event showing a birthday pair, it increases the counter in `matches` by one. At the end of each run, the script prints the fraction of the number of parties with shared birthdays relative to the total number of parties; Fig-

Listing 2: bp-sim

```
01 #!/usr/bin/env python3
02
03 import random
04
05 def bday_match(bdays):
06   seen = set()
07   for bday in bdays:
08     if bday in seen:
09       return True
10     seen.add(bday)
11   return False
12
13 for epoch in range(10):
14   parties    = 100000
15   matches    = 0
16   nof_days   = 365
17   nof_guests = 23
18
19   for party in range(parties+1):
20     guest_bdays=[]
21     for _ in range(nof_guests):
22       bday = random.randint(1,nof_days)
23       guest_bdays.append(bday)
24
25     if bday_match(guest_bdays):
26       matches += 1
27
28   print(matches/parties)
```

ure 2 shows that the value settles at about 50.7 percent.

The `bday_match()` function from line 5 expects a Python list with integers and checks if there are one or more duplicates. This test is efficient because it uses a hash function to squash previously seen values into a `seen` set; it can then quickly check whether the value is already in the set for each newly examined value. If you have ever had this task in a recruitment test, you will be aware that the compute time for the duplicate check using this procedure drops to O(n) for n list elements, while it would be O(n^2) for a less clever two-loop solution.

Black on White

How does the probability of a birthday collision develop with an increasing number of guests? Thanks to the `matplotlib` Python library, simply installed with

```
pip3 install -user matplotlib
```

```
$ ./birthday-paradox
 1: 0.0000
 2: 0.0027
 3: 0.0082
 4: 0.0164
 5: 0.0271
 6: 0.0405
 7: 0.0562
 8: 0.0743
 9: 0.0946
10: 0.1169
11: 0.1411
12: 0.1670
13: 0.1944
14: 0.2231
15: 0.2529
16: 0.2836
17: 0.3150
18: 0.3469
19: 0.3791
20: 0.4114
21: 0.4437
22: 0.4757
23: 0.5073
```

Figure 1: Probability of a double birthday.

```
$ ./bp-sim
0.50763
0.50517
0.50695
0.5069
0.50967
0.50604
0.50765
0.50664
0.50894
0.50634
```

Figure 2: The 23-person simulation script settles at about 50 percent probability of a match.

Listing 3: bd-plot

```
01 #!/usr/bin/env python3
02
03 import matplotlib.pyplot as plt
04 import sys
05
06 x=[]
07 y=[]
08 for line in sys.stdin:
09   (guests,prob)=line.split(': ')
10   x.append(guests)
11   y.append(prob)
12
13 plt.plot(x,y)
14 plt.xlabel('Guests')
15 plt.ylabel('Probability')
16
17 plt.savefig('bd-collision.png')
```

Listing 3 produces the graph in Figure 3 with the output data from Listing 1:

With the special file handle sys.stdin, Listing 3 reads the output lines of Listing 1 and uses the split() method in line 9 to split them at the colon, thus separating the number of guests and the probability. For the x and y values in the graph, it compiles the x and y lists by appending the latest value with the append() method to the respective list for each value pair. The plot() method then collectively accepts all x and y values and draws the graph that the savefig() method writes to a PNG image file in line 17. It can hardly be done with less effort, and the graph looks quite appealing.

Figure 3: Probability of equal birthdays for more guests.

```
./birthday-paradox | ./bd-plot
```

What happens at a party with 100 participants can also be determined with these scripts: The probability of two guests sharing a birthday bounces up to 99.99996928 percent; the chance that all guests have different birthdays at this mega party is more than 1 in 3 million. ∎∎∎
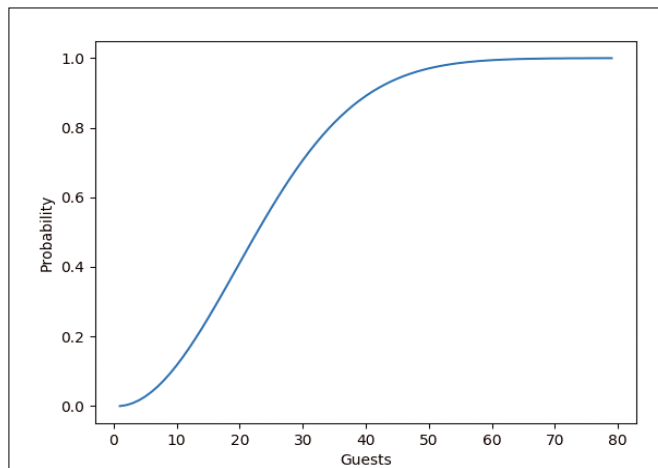
Info

[1] The Birthday Problem/Paradox: https://www.youtube.com/watch?v=QrwV6fJKBi8

[2] Listings for this article: ftp://ftp.linux-magazine.com/pub/listings/linux-magazine.com/211/
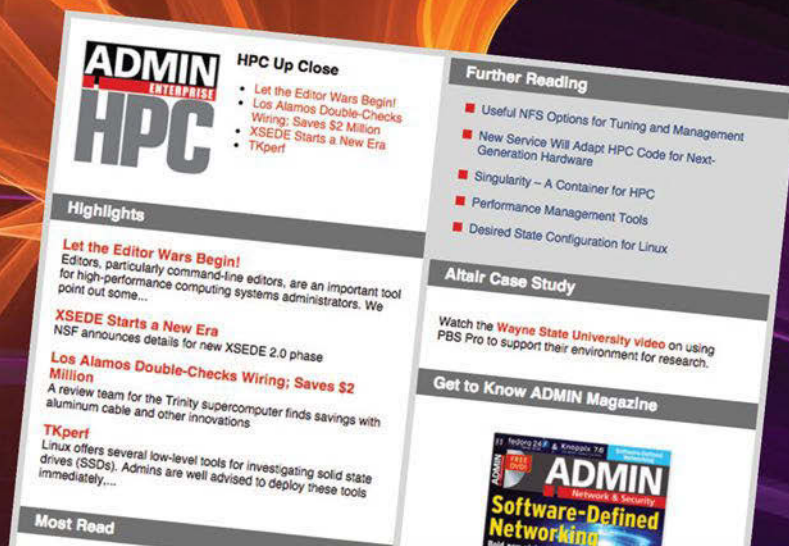
Open Source Backup CONFERENCE

September 26, 2018 | Cologne

CALL FOR PAPERS & EARLY BIRD
until June 15

OSBCONF.ORG

FOCUSSED ON

bareos
open source data protection

ReaR

Amanda

Meltdown, Spectre, and what they mean for Linux users

SUPERBUGS

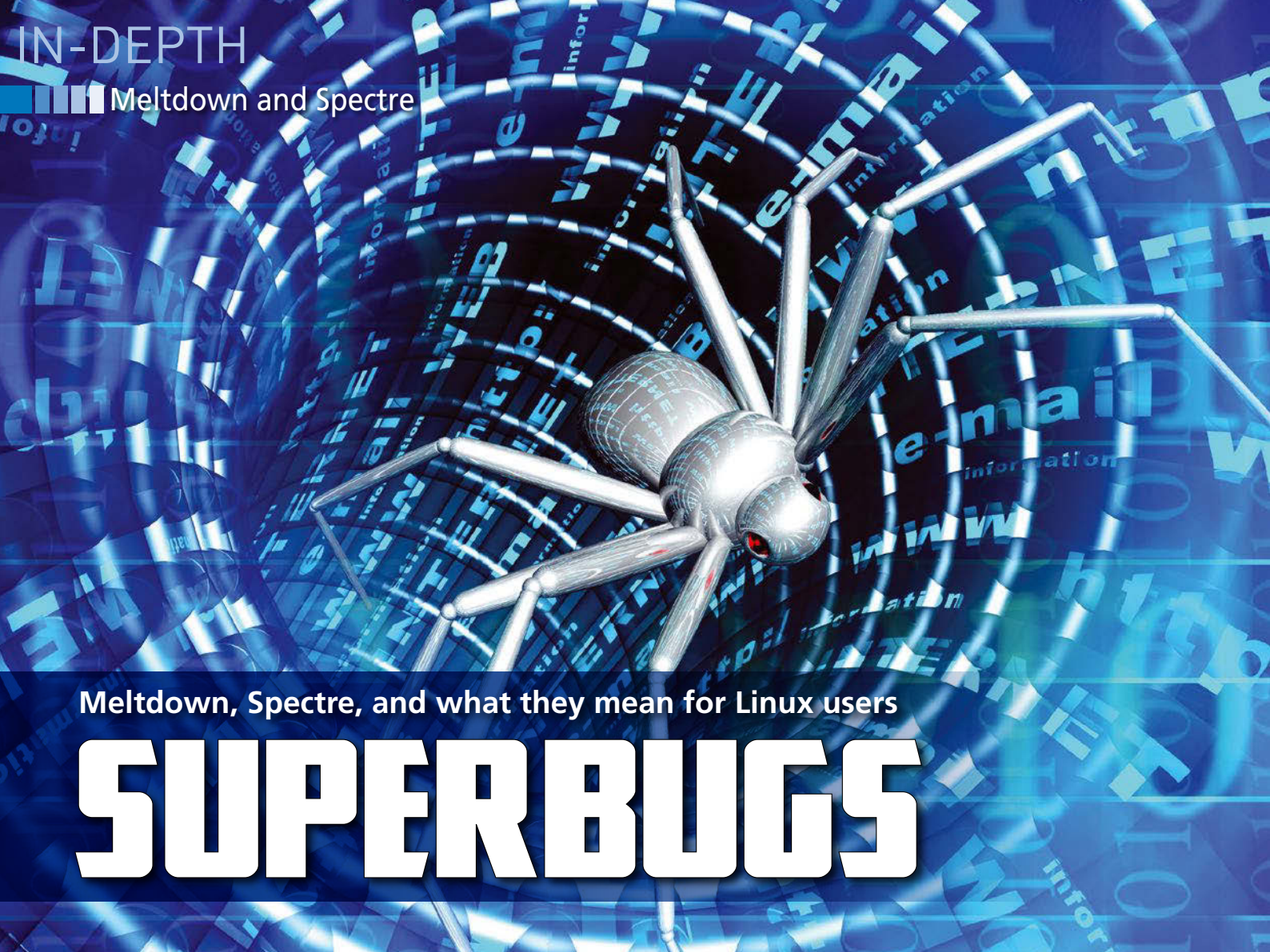The blatant security holes known as Meltdown and Spectre, which are built into the computer hardware, are likely to keep us busy for the next few years. How is the Linux community addressing this unexpected challenge? *By Ferdinand Thommes*

The year 2018 began with a disaster for IT: We learned that most processors sold in the last 15 years come with two blatant bugs that make our systems vulnerable. The vulnerabilities, named Meltdown and Spectre, mainly affect CPUs of the market leader Intel, but related problems are also present in Apple, AMD, PowerPC, and ARM64 processors. (To the relief of makers around the world, the all-clear has been issued for all Raspberry Pi models.)

These security gaps, which are largely a result of the race for increasingly faster computers, will persist for a very long time and can only be completely eliminated with a new generation of CPUs – probably years in the future. Kernel developers will have to deal with the vulnerabilities that are opening up on PCs, smartphones, and even cloud service for a long time to come. With smartphones and tablets, only owners of currently supported devices can hope to eliminate the vulnerabilities; older devices remain unprotected.

In addition to the vulnerabilities that became known in January, security researchers published further attack scenarios [1] on February 14th. The previous software patches probably also cover these new attack vectors, but for Intel, this means that the changes to the CPU blueprints developed so far must be scrapped, and the engineers have to go back to the drawing board.

Intel Stonewalling

In the Spring and Summer of 2017, two research teams at Google and the University of Graz found alarming indications that volatile data such as passwords and other confidential data can be read from the memory of almost all current CPUs. The procedure requires in-depth knowledge and suitable tools. Potential perpetrators are therefore primarily government organizations and well-positioned cybercriminals.

Intel, ARM, and AMD were first informed about the gaps by Google researchers on June 1, 2017. It seems logical that a company first has to check such information internally. However, the fact that Intel sold faulty CPUs for half a year without informing customers is highly questionable and is already the basis for more than 30 individual and collective lawsuits in the USA.

In other respects also, Intel repeatedly gives the impression that the company is

Lead Image © Natalia Lukiyanova, 123RF.com

more interested in the share value than in the customers. The fact that Intel CEO Brian Krzanich sold a good portion of his 495,000 Intel shares well in advance of the announcement of the gap seems particularly sensitive. He only kept the shareholding that he had to have, according to the company regulations, in order not to lose his job.

So much for the bare facts. Home users can take comfort in the fact that attackers can only exploit these vulnerabilities at great expense. But that doesn't make things any better, especially given the inglorious approach of the lawyers and business economists who were determining the company's reaction. The strategy seemed to be first keep quiet, then deny for as long as possible, and finally fend off complaints if denial is no longer possible.

At the same time, the company had its developers throw together new microcode [2], which it then had to withdraw promptly. For Linux, the main burden of containment fell to the kernel developers. In the middle of the Linux 4.15 development cycle over the holidays at the end of the year, they were surprised by the dilemma that, in contrast to what Intel had intended, became public before 9th January. The kernel maintainers then erupted into hectic activity as they tried to mitigate the consequences of a catastrophe for which they themselves were not responsible. This effort pushed some developers to the brink of their capacity.

Linus Torvalds, master of the kernel, had sharp words for Intel's behavior and the company's unsuitable repair attempts [3]: He called Intel's hasty repair attempts "absolute rubbish" and "completely insane." Kernel developer Greg Kroah-Hartman stressed that Intel's repair attempts were a prime example of how not to deal with the kernel development community. The gaps even caused a new virtual directory to appear in the

kernel tree at `/sys/devices/system/cpu/vulnerabilities/` (Figure 1).

Speculative Execution

Meltdown and Spectre attack scenarios exploit certain mechanisms in modern processor architectures. They are particularly aimed at out-of-order execution [4], which accelerates modern processors (Figure 2). The technique speculatively executes instructions, so that the processor has already loaded the required instruction when it needs it and does not have to wait to painstakingly access memory.

In some cases, the existence of this speculative data allows a capable attacker to access the kernel memory from user space. In this way, an attacker can potentially access all data that the machine is currently storing in memory, including passwords, security certificates, and other security-relevant data.

Out-of-order execution changes the system state at certain points, even if the CPU rejects the result of speculative execution. With appropriate software, the attacker can thus cause the attacked process to speculatively execute injected instructions.

The vulnerabilities have given rise to three different attack scenarios, known as Meltdown, Spectre v1, and Spectre v2. In general, because of actions taken by the kernel developers, Linux users were protected earlier and in a better way than Windows users. Linux users also have an edge when it comes to countermeasures taken by chip manufacturers: In many distributions, you directly receive the patches embedded in microcode as installable packages. Windows users have to wait far longer because the manufacturers

first hand over the patches to the OEMs, who then have to adapt them to the hardware; only then can they pass the changes on to the customers as BIOS or firmware updates.

The microcode installed under Linux as a package is loaded each time the computer is started, which removes the need for a BIOS update on Linux computers. However, the microcode itself is proprietary software, so you have to unlock the *contrib* and *nonfree* repositories if you are installing on an all-free distro such as Debian.

Clarified

An Intel paper from February 11, 2018 provides clarity for the first time regarding which processors will need new microcode and what is the current state of development [5]. All information in this document refers to Spectre v2. Closing this gap requires a kernel patch as well as new microcode from Intel. In the paper, Intel for the first time also lists older CPUs that were launched about ten years ago. Ivy Bridge, Sandy Bridge, Penryn, Wolfdale, and Yorkfield will now also be receiving microcode updates.

Intel released a new microcode version in early February, which was initially only stable for Skylake processors. At the time this article was written, microcode had not yet reached the Linux distributions; according to Intel, versions for further processors were still in beta testing at that time.

The fundamental difference between Spectre and Meltdown is that Spectre manipulates a process such that the process reveals its own data, whereas Melt-

```
# cd /sys/devices/system/cpu/vulnerabilities/
# ls
meltdown   spectre_v1   spectre_v2
# cat meltdown
Mitigation: PTI
# cat spectre_v1
Mitigation: __user pointer sanitization
# cat spectre_v2
Mitigation: Full generic retpoline
```

Figure 1: A new virtual directory lists the measures taken against the Meltdown and Spectre attacks.
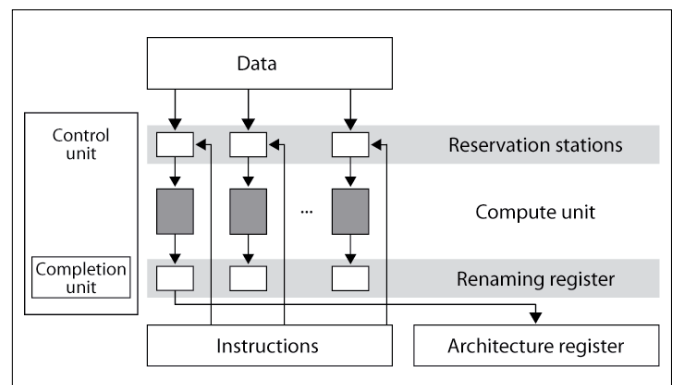
Figure 2: In the case of out-of-order command execution, the CPU speculatively executes several possible commands, one of which it actually implements, while it discards the others. (Image: Sputniktilt/Michael Kleine, commons.wikimedia.org, CC-BY-SA 2.0)

down can read privileged memory in the address space of a process that even the process itself normally cannot access.

Meltdown

Meltdown [6] breaks the security barriers between userspace programs and the kernel. A process prepared by the attacker can cause the CPU to speculatively load data and then access the kernel area directly by levering out access control – for example, using untrusted JavaScript code executed via a web browser.

The problem is that many CPUs do not check access rights when accessing memory during speculative execution. Attackers can therefore load data into the cache that should not actually be accessible.

There is no direct way to read the data from the cache; however, differences in the execution times of data in the cache and out of it can be used to read individual bits in memory. By repeating the procedure, malicious code can read the memory at a rate of up to 1.5KBps – fast enough to access passwords and other security-relevant data stored there.

To prevent this attack, the developers of Kernel 4.15 and higher separate the page tables formerly shared by the kernel and user space into two completely separate sets. Thanks to kernel page table isolation (KPTI), after you apply the appropriate patches, a process can only see the memory area assigned to it; no longer can it see the memory of the operating system

kernel [7]. This restriction prevents an unprivileged process from accessing the kernel space memory area.

AMD in Luck

The Meltdown gap has plagued almost all Intel processors since 1995, with the exception of the Itanium architecture, the Atom processors produced before 2013, and some ARM64 chips based on the Cortex A75. AMD CPUs, on the other hand, are not affected by Meltdown. The Linux kernel already benefited from Meltdown patches at the end of the year, so that protection has existed as of Linux 4.15-rc7, from January 6.

The developers then backported the fix to the LTS 4.4 and 4.9 kernels, as well as Linux 4.14.12; Kernel 4.16 protects affected ARM64 CPUs and the s390 port for Linux on IBMs Z-systems and LinuxONE servers. The 3.18, 4.4, and 4.9 branches of the Android Common Kernel were patched. Kernel 4.15 offers fundamental protection against Meltdown, although it is still vulnerable to Spectre v1 (Figure 3). Kernel 4.15.4 from February 16 closes the Spectre v1 vulnerability (Figure 4). This is by no means the end, as 4.17 and beyond will further stabilize and expand the existing patches.

Spectre

Defensive measures against Spectre [8] cause considerably more work than defenses that target Meltdown. Spectre also affects AMD

Figure 3: The Spectre and Meltdown mitigation and detection tool shows Spectre v1 as a vulnerability with Kernel 4.15.0. Meltdown and Spectre v2 are closed.

processors and the PowerPC platform. Initial patches for Spectre v2 were available in kernel 4.15; they are successively extended and perfected in Kernel 4.16.

The two variants of the Spectre vulnerability can only be prevented by the interaction of a patched kernel, an adapted compiler, and microcode from Intel or AMD. In addition, many thousands of applications have to be patched separately; Firefox 57.0.4, Chrome 64, and other web browsers already have updates.

Linux uses a software construct from Google to combat Spectre, which the developers named Retpoline ("Return Trampoline") [9]. This fix allows indirect branches to be excluded from speculative execution via an endless loop. Retpoline needs an adapted compiler, which is now available in the form of GCC 7.3.

At the same time, Intel released new microcode to contain Spectre v2 with the Indirect Branch Prediction Barrier (IBPB), Single Thread Indirect Branch Predictors (STIBP), and Indirect Branch Restricted Speculation (IBRS)

measures. These changes are what unleashed Linus Torvalds' criticism. He expressed the opinion that the changes were "absolute rubbish" and a "dirty hack" that would result in severe performance loss. Intel indirectly proved him right by hastily withdrawing the microcode and discouraging its use. The updates led to crashes and spontaneous reboots on Broadwell and Haswell CPUs.

The kernel community, on the other hand, relies on Google's Retpoline fix, which does not compromise performance. The only potential addition to the Linux kernel fix is Intel's IBPB, which serves as a barrier for context changes to prevent the CPU from reusing already known branch targets. However, before the kernel community can integrate IBPB, CPU manufacturers must first make revised microcode available.

The last of the three vulnerabilities the kernel developers need to patch is Spectre v1. On February 4, the first basic patches designed to contain Spectre v1 were submitted; they have already been included in Kernel 4.15.4.

Further improvements will follow for kernel 4.16, expected at the beginning of April, and later kernel versions. These code patches affect both the x86 platform and ARM64.

Performance

The measures for containing Meltdown and Spectre reduce the computing power of all affected processors, with the measures that target Meltdown leading to the biggest declines. Performance losses for desktop computers usually do not exceed five percent, but servers can lose 30 percent or more, depending on the usage profile.

The effects depend strongly on the number of system calls and the context change between user space and kernel due to page table isolation. Both Intel and the kernel developers expect to reduce the performance impact in the near future.

Conclusions

Linux users are the most up to date when it comes to Meltdown and Spectre: Either Linux 4.16.x or the back-patched
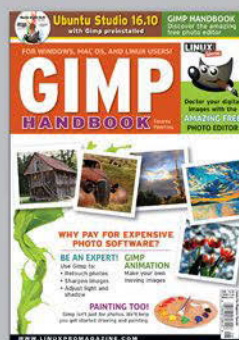
LTS kernels of the 4.4, 4.9, and 4.14 series are the benchmark. Users should ditch any distribution that is not kept up to date. In contrast, Windows users are in the dark, as all protections are delivered to them with a time delay. BSD was given its first patches on February 17; the BSD patches resemble those for Linux.

The fundamental problem lies in the silicon of the processors and can only be solved there. Even a new *stepping level* revision of a CPU takes up to three years and costs millions. (A *stepping* is a sub-version of a processor generation, corresponding to a minor version change for software – also known as a core revision at AMD.) It is also questionable whether all necessary countermeasures could be accommodated in a single stepping. Another difficulty is fixing the vulnerabilities without creating new bugs.

Cautious estimates assume that the Meltdown and Spectre gaps could still be lurking in the background for a decade. The new MeltdownPrime and SpectrePrime variations announced in February require further changes to the silicon – and they are unlikely to be the last.

So far, it is unlikely that any malware in the wild can actually exploit Meltdown and Spectre. But any delays in implementation are a reflection on the complexity of the attack vectors – not the interest of the attackers. The first, fortunately still ineffective, malware that exploits the gaps is already appearing.

The most complete information on Meltdown and Spectre was published by the discoverers of the vulnerabilities in Google's Project Zero [10]. You can check the current state of protection for your own hardware with a script that you download from GitHub [11] and run as root (Listing 1). Under the latest Debian and Ubuntu versions, install the script directly from the repositories using the *spectre-meltdown-checker* package. ∎∎∎

```
Spectre and Meltdown mitigation detection tool v0.36

Checking for vulnerabilities on current system
Kernel is Linux 4.16.0-towo.1-siduction-amd64 #1 SMP PREEMPT siduction 4.16-1 (2018-04-03) x86_64
CPU is Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz

Hardware check
* Hardware support (CPU microcode) for mitigation techniques
  * Indirect Branch Restricted Speculation (IBRS)
    * SPEC_CTRL MSR is available:  YES
    * CPU indicates IBRS capability:  YES  (SPEC_CTRL feature bit)
  * Indirect Branch Prediction Barrier (IBPB)
    * PRED_CMD MSR is available:  YES
    * CPU indicates IBPB capability:  YES  (SPEC_CTRL feature bit)
  * Single Thread Indirect Branch Predictors (STIBP)
    * SPEC_CTRL MSR is available:  YES
    * CPU indicates STIBP capability:  YES
  * Enhanced IBRS (IBRS_ALL)
    * CPU indicates ARCH_CAPABILITIES MSR availability:  NO
    * ARCH_CAPABILITIES MSR advertises IBRS_ALL capability:  NO
  * CPU explicitly indicates not being vulnerable to Meltdown (RDCL_NO):  NO
  * CPU microcode is known to cause stability problems:  NO  (model 94 stepping 3 ucode 0xc2)
* CPU vulnerability to the three speculative execution attack variants
  * Vulnerable to Variant 1:  YES
  * Vulnerable to Variant 2:  YES
  * Vulnerable to Variant 3:  YES

CVE-2017-5753 [bounds check bypass] aka 'Spectre Variant 1'
* Mitigated according to the /sys interface:  YES  (kernel confirms that the mitigation is active)
* Kernel has array_index_mask_nospec:  YES  (1 occurence(s) found of 64 bits array_index_mask_nospec())
* Kernel has the Red Hat/Ubuntu patch:  NO
> STATUS:  NOT VULNERABLE  (Mitigation: __user pointer sanitization)

CVE-2017-5715 [branch target injection] aka 'Spectre Variant 2'
* Mitigated according to the /sys interface:  YES  (kernel confirms that the mitigation is active)
* Mitigation 1
  * Kernel is compiled with IBRS/IBPB support:  YES
  * Currently enabled features
    * IBRS enabled for Kernel space:  UNKNOWN
    * IBRS enabled for User space:  UNKNOWN
    * IBPB enabled:  UNKNOWN
* Mitigation 2
  * Kernel compiled with retpoline option:  YES
  * Kernel compiled with a retpoline-aware compiler:  YES  (kernel reports full retpoline compilation)
> STATUS:  NOT VULNERABLE  (Mitigation: Full generic retpoline, IBPB, IBRS_FW)

CVE-2017-5754 [rogue data cache load] aka 'Meltdown' aka 'Variant 3'
* Mitigated according to the /sys interface:  YES  (kernel confirms that the mitigation is active)
* Kernel supports Page Table Isolation (PTI):  YES
* PTI enabled and active:  YES
* Running as a Xen PV DomU:  NO
> STATUS:  NOT VULNERABLE  (Mitigation: PTI)
```

Figure 4: With Kernel 4.16, the Spectre v1 vulnerability is now mitigated. The patches will be further expanded and cleaned up with 4.17.

Info

[1] New attack vectors:
https://www.zdnet.com/article/meltdown-spectre-flaws-weve-found-new-attack-variants-say-researchers/

[2] Microcodes: https://en.wikipedia.org/wiki/Microcode

[3] Torvalds' comments:
http://lkml.iu.edu/hypermail/linux/kernel/1801.2/04628.html

[4] Out-of-order execution:
https://en.wikipedia.org/wiki/Out-of-order_execution

[5] Microcode (version 11.02.):
https://newsroom.intel.com/wp-content/uploads/sites/11/2018/02/microcode-update-guidance.pdf

[6] Meltdown: https://en.wikipedia.org/wiki/Meltdown_(security_vulnerability)

[7] KPTI: https://lwn.net/Articles/741878/

[8] Spectre:
https://en.wikipedia.org/wiki/Spectre_(security_vulnerability)

[9] Retpoline: https://support.google.com/faqs/answer/7625886

[10] Project Zero: https://googleprojectzero.blogspot.de/2018/01/reading-privileged-memory-with-side.html

[11] spectre-meltdown-checker:
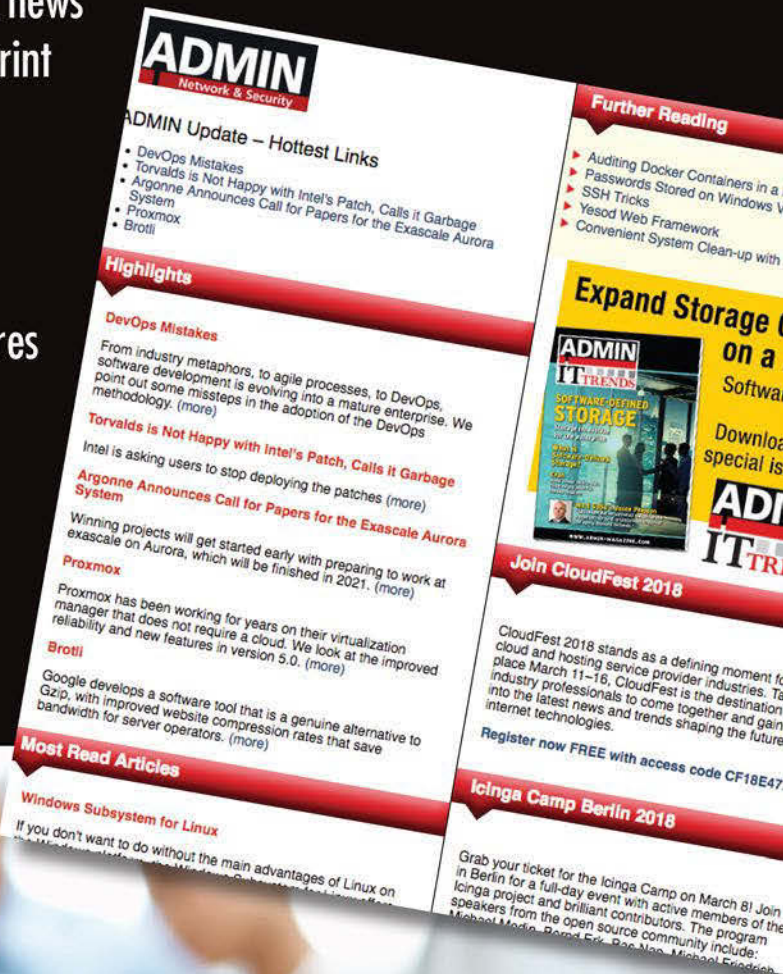https://github.com/speed47/spectre-meltdown-checker

Listing 1: Vulnerability Check

```
$ wget meltdown.ovh -O spectre-meltdown-check

$ chmod u+x spectre-meltdown-checker.sh

$ sudo ./spectre-meltdown-checker.sh
```

A Python solution for a classic chess puzzle

Knight's Tour

If you're looking for a head start on solving the classic Knight's Tour chess challenge, try this homegrown Python script. *By Gordon Horsington*

A chess knight can travel around a chessboard using the knight's move and visit every square once, and only once. This is known as the Knight's Tour. The Knight's Tour is usually played with numbered counters on a conventional chessboard (Figure 1), but you can play it on any rectangular board with five or more rows and columns.

The Knight's Tour is an example of a classic mathematical problem that lends itself to easy and creative expression through computer programming. I created a Python program to help users practice solving the Knight's Tour. The Knight's Tour program is a good example of the simple yet powerful applications you can build with the Pygame Python library for computer gamers.

Author

Gordon doesn't really like the sort of confrontational competition that playing chess seems to encourage, and which inevitably creates winners and losers. As he's both a bad winner and a bad loser, Gordon prefers chess puzzles to actually playing chess.

Solving the Knight's Tour

At first, solving the Knight's Tour seems to be a daunting challenge, but quite a few strategies exist that will help you solve the puzzle [1]. I'll consider just two of them. The first strategy is usually known as the Diamonds and Squares method, and it's a simple party trick that anyone can learn in minutes.

Solving the Knight's Tour using the Diamonds and Squares technique doesn't require any mathematical talent

Figure 1: The Knight's Tour is usually played on a chessboard with counters to mark the progress. This tour starts on the square with the black knight and ends on the square with the white knight.

or much logical skill; it merely requires the player to recognize two simple geometric shapes in the patterns of knight's moves. This approach can produce a relatively small number of apparently unique tours from every starting square, but the symmetry of the chessboard means that many of these solutions are transformations of other tours by rotation, reflection, or inversion, and this simple technique can only produce 46 truly unique Knight's Tours on a chessboard. Most of the Knight's Tour solution videos you watch on YouTube and elsewhere demonstrate the Diamonds and Squares method.

The second approach is known as Warnsdorff's rule. This strategy takes a bit more effort to master. On an 8x8 chessboard, it can produce thousands of different Knight's Tours from any starting square, but, again, the symmetry of the chessboard means that not all of these solutions are truly unique and many of them are transformations of other Warnsdorff's rule tours. As you'll see in a moment, Warnsdorff's rule can sometimes lead to an impasse, but it still produces far more solutions than the Diamonds and Squares method.

If you want to impress your friends by showing them how to solve the Knight's Tour, then using Warnsdorff's rule will not expose you as a chess puzzle pretender, but using the simple Diamonds and Squares technique might put your reputation as a chess puzzle genius in jeopardy.
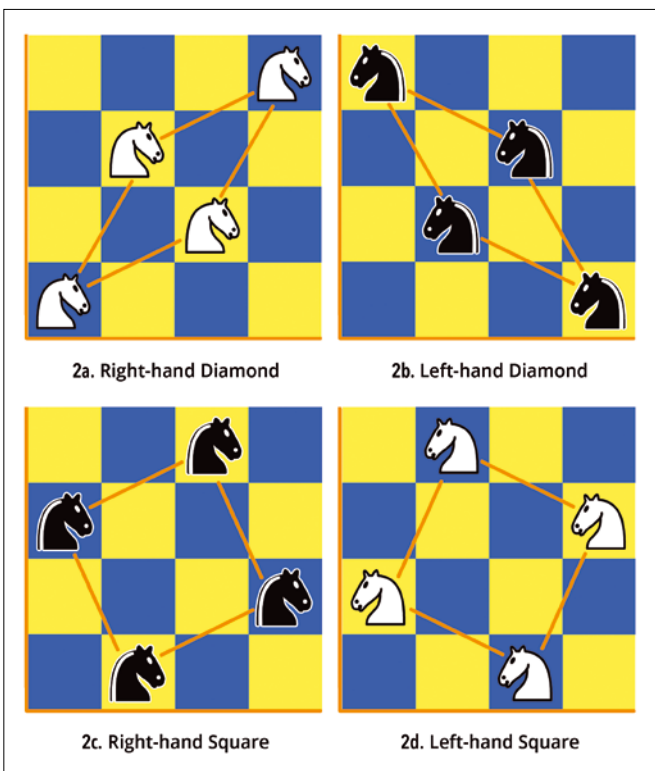
Diamonds and Squares

The Diamonds and Squares technique was first described by C. R. R. von Schinnern in 1826 [1]. It only works on boards that have at least 8x8 squares. Larger boards must have a number of rows and columns that are a multiple of four (i.e., 12x12, 8x16, and so on). Unlike Warnsdorff's rule, the Diamonds and Squares technique cannot be used to solve the Knight's Tour on a 6x6 or a 10x10 board.

To employ the Diamonds and Squares strategy on a conventional 8x8 chessboard, you need to divide the chessboard into four quadrants and understand how four knights can be placed on a quadrant in four distinct patterns. The first of these patterns is described as a right-hand (RH) diamond (Figure 2a). The second is a left-hand (LH) diamond

(Figure 2b), the third is a RH square (Figure 2c), and the last one is a LH square (Figure 2d). When these four shapes are joined together on one quadrant of the chessboard, all 16 squares in that quadrant will have a knight on them.

In order to use the diamond and square shapes to solve the Knight's Tour, you need to know whether the first knight placed on the board is on a diamond (either RH or LH) or on a square (either RH or LH). For example, a knight at b3 is on a RH diamond (Figure 3a), one at c4 is on a RH square (Figure 3b), a knight at d1 is on a LH diamond (Figure 3c), and one at d3 is on a LH square (Figure 3d).

If you place the first knight on, for example, b3, then you can complete a four-knight partial tour in the lower left-hand quadrant with four knights on a RH diamond shape (Figure 3a). You then move onto an adjacent quadrant and use a RH diamond shape to complete another four-knight partial tour on that quadrant. Then you can move on to a third quadrant and then the fourth so that all four quadrants have a four-knight, RH diamond-shaped partial tour

Figure 2: Any chessboard quadrant can be filled with 16 knights arranged in four shapes: 2a is the right-hand diamond, 2b the left-hand diamond, 2c the right-hand square, and 2d the left-hand square.
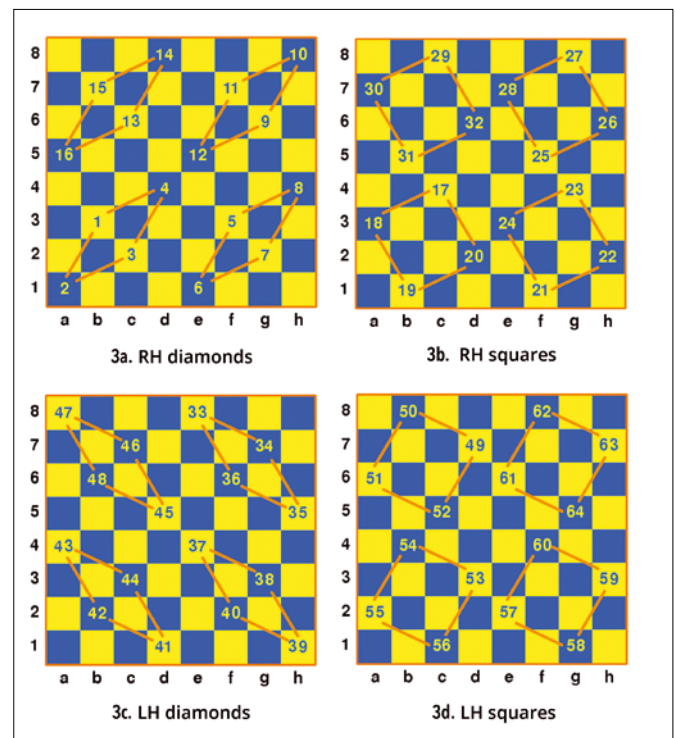
Figure 3: The Knight's Tour can be completed by combining four, 16-knight, partial tours using all four shapes; in this example, RH diamonds are followed by RH squares, and then LH diamonds are followed by LH squares.

on them, and there are 16 knights on the board (Figure 3a). After completing the RH diamond-shaped partial tour, you move on to a square-shaped partial tour, in this example, a RH square-shaped partial tour starting at c4 and ending at d6 (Figure 3b). Then you can complete a LH diamond-shaped partial tour starting at e8 and ending at b6 (Figure 3c) and finally a LH square-shaped partial tour starting at d7 and ending at g5 to complete the Knight's Tour (Figure 3d).

The Knight's Tour can always be solved by combining two, 16-knight, diamond-shaped, LH and RH partial tours and two, 16-knight, square-shaped, LH and RH partial tours in the order diamond-square-diamond-square or square-diamond-square-diamond. You will need to make sure that the fourth knight placed in each quadrant leads to an empty square on an adjacent quadrant. With a bit of practice, it's quite easy to solve the Knight's Tour from any starting square using this method.

You can find variations of the Diamonds and Squares technique that will allow you to choose the starting and ending square, but be warned, nearly everyone who knows about the Knight's Tour knows about the Diamonds and Squares method, and you can only impress naive chess puzzlers by demonstrating this technique.

Warnsdorff's Rule

The second strategy is known as Warnsdorff's rule [1, 2] and it is named after H. C. von Warnsdorff, who first described it in 1823. Warnsdorff's rule is a heuristic or rule-of-thumb method for finding a Knight's Tour from any starting square on any sized board with at least 5x5 squares. I'll look at it on a conventional chessboard.

Warnsdorff's rule is summarized as follows:

1. With each move, you have to look ahead to see which of the possible next moves has the least number of exits that can be taken using the knight's move (Figure 4). An exit is a legal move to a square that the knight hasn't inhabited yet.
2. The square with the least number of valid exits is chosen for the next move.

3. If there is a tie between two or more squares, the next square is chosen at random from all the squares with the least number of exits.

Unfortunately, if a Monte Carlo test [3] is done when Warnsdorff's rule is applied to a conventional chessboard, about three percent of the attempts to find a complete tour will lead to an impasse. If you want to find a Knight's Tour every time using Warnsdorff's rule, you need to modify the original rule to overcome any impasse that might occur. Do this by replacing the random choice of a tiebreaker with a systematic choice.

The easiest variant of Warnsdorff's rule to implement in a computer program requires very little coding. It modifies the third element of Warnsdorff's rule so that, if there is a tie for the next move between two or more squares, the next square is chosen systematically – any systematic way of choosing the tiebreaker will do as long as it prioritizes the way in which the tied moves are selected to break the tie. If breaking a tie systematically leads to an impasse, the calculation can just start again using a different way of making a systematic choice. This is quite a good option for a computer program, because a computer can find a complete tour using an alternative way of choosing a tiebreaker in milliseconds. The program overcomes any impasse that might occur using this simple plan – if the way you're choosing a tiebreaker doesn't work, then stop using it, start again, and choose your tiebreaker in a different way.

4a. Five moves from b5

4b. Seven moves from c4

4c. Five moves from c2

4d. Two moves from b1
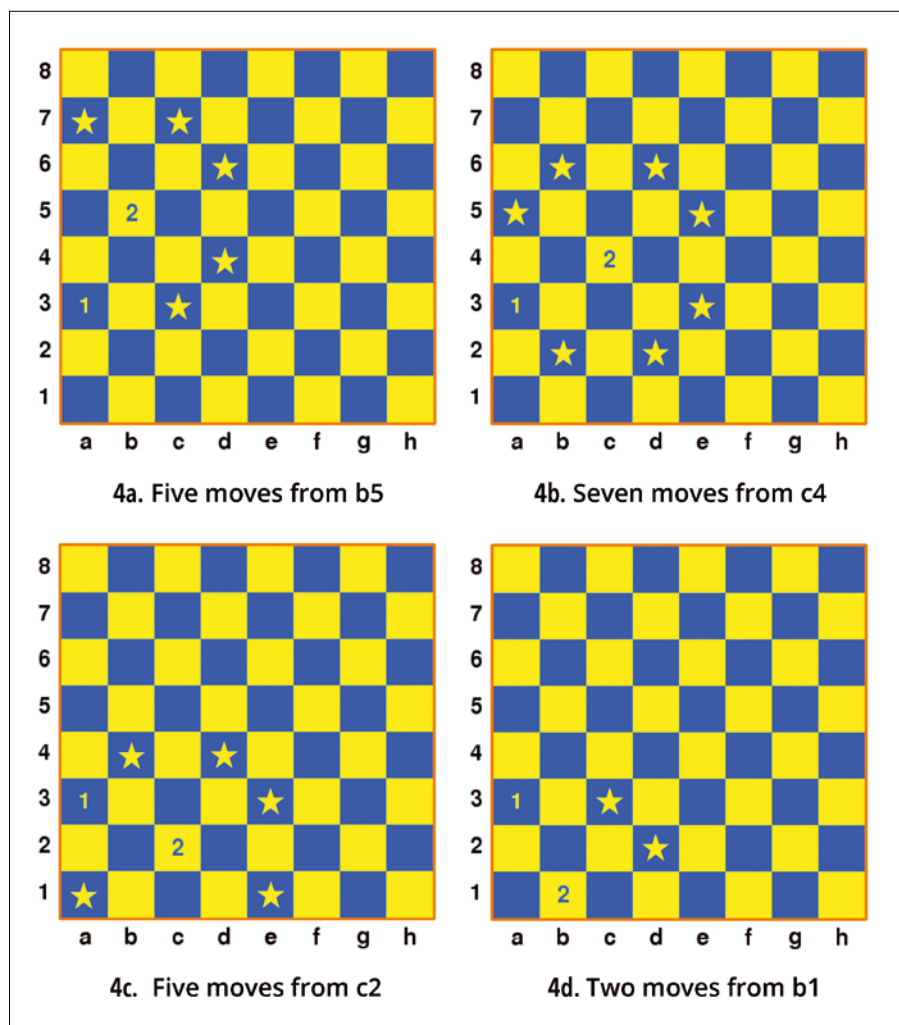
Figure 4: Warnsdorff's rule requires that you look ahead to find the next move that has the least number of exits. In this example, looking ahead from the first position at a3 shows that a move to the second position at b1 has the least number of exits (the exits are marked with stars).

Another variant of Warnsdorff's rule is better if you're trying to find a way out of an impasse by hand, because it usually involves less work than starting again. If systematic tie breaking leads to an impasse, you can backtrack to the last tiebreaker and take the second-priority move. If that doesn't work, you can backtrack again and choose the third-priority move, and so on until all the tiebreakers have been tried. If that doesn't overcome the impasse, backtrack to the last-but-one tiebreaker, select the second priority tiebreaker, and keep doing this until the impasse is overcome. The computer program will enable you to backtrack in this way if you're using Warnsdorff's rule (or any other strategy) to solve the puzzle by hand and you get stuck in an impasse. The program doesn't use backtracking when it's finding a solution for you, but if you'd like to see how a program can use backtracking to find a way out of an impasse, there's a backtracking algorithm used in a Knight's Tour program I wrote when I first became interested in chess puzzles many years ago. This program was published in *The Century/Acorn User Book of Computer Puzzles* [4].

Any systematic way of choosing can be used as a tiebreaker for either of the alternative versions of Warnsdorff's rule described above, but remember, it must be a systematic choice rather than a random one. There are eight factorial (*8! = 40320*) ways in which you can make a systematic choice, but two of the easiest ones to remember when finding a Knight's Tour by hand are to select the tiebreaker in either a clockwise or counterclockwise direction starting at the 12 o'clock position (i.e., start looking from the top of the board and work your way around the tied moves to the right or to the left).

A very handy shortcut version of Warnsdorff's rule can be used to place the first 30 or so knights on the board. This technique works because, in the first half of solving the puzzle, selecting the square with the least number of exits is usually the same as moving the knight as far away from the center of the board as possible – but beware, this shortcut will occasionally disappoint you.

The shortcut requires you to stay as close to the edge of the board as possible and to prioritize the corner squares and their adjacent perimeter squares. This will usually result in two circuits of the board's perimeter during the first half of the tour, after which it's better to stop using the shortcut and start calculating the best way around the empty squares on the board using systematic tie breaking. An insight into what's needed to solve the puzzle is also very useful during the second half of the tour, and you will develop the necessary insight with practice. The fewer knights that are placed on the board using this shortcut, the better it works. The shortcut becomes less reliable as more knights are placed on the board, and the chance that it will let you down increases noticeably when there are more than half of the knights on the board. The Knight's Tour shown in Figure 5 was found using this shortcut combined with backtracking and insight. With practice, it takes only a couple of minutes to find a complete Warnsdorff's tour by hand.

Knight's Tour Program

The Knights Tour program will enable you to practice solving the Knight's Tour by hand using the Diamonds and Squares method, Warnsdorff's rule, or any of the many other strategies [1, 2]. The program (`knights_tour.py`) is available for download at the *Linux Pro Magazine* website [5]. To use the program, you'll need Python 3 and the Pygame library [6], which is available as a package on many Linux systems. You can also use the Knight's Tour program to find and display a huge number of solutions from any starting square. A mouse is used for interacting with the program. You can click on any of the options on the right-hand side of the window and click on any of the chessboard squares. On the right-hand side of the window are five options from which to choose (Figure 5):

1. *leave this game* is self explanatory.
2. *start/restart game* is selected by default when the program starts. This option allows you to choose the starting square for the puzzle and then to place the next and subsequent knights on the board. You can backtrack while this option is active and restart the game by clicking on this option at any time.
3. When the *next move on/off* option is *on*, it places a small star in the top-left corner of squares that are legitimate knight's moves to indicate where the next move can be made when you're attempting to solve the puzzle. This includes a backtracking square – very useful if you want to practice Warnsdorff's rule. You can
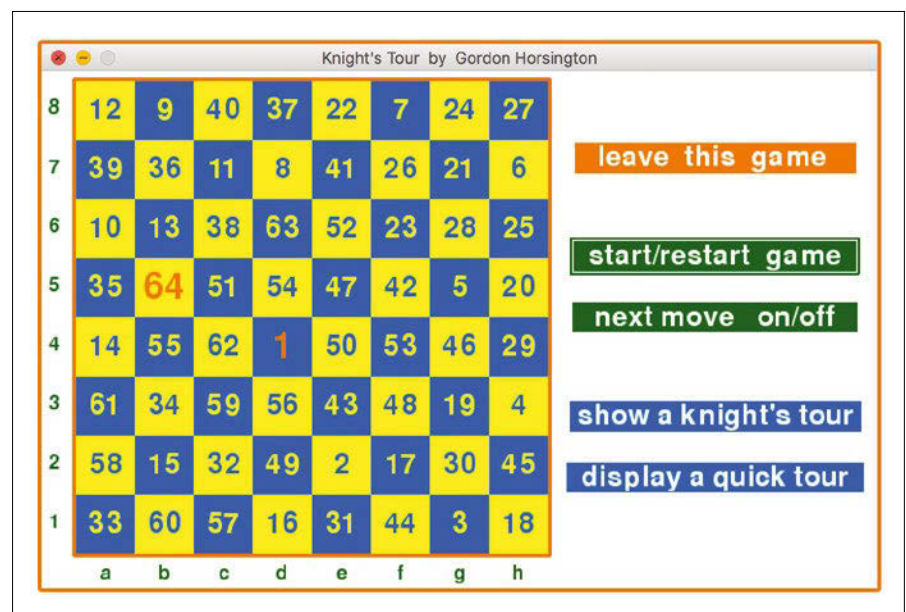
Figure 5: The Knight's Tour program will help you to solve the puzzle using any strategy, including backtracking if you choose. The tour in this illustration is described as a closed tour, because the first and last positions are just one knight's move apart. The program will solve the puzzle for you if you click on one of the blue options on the right and then select a starting square.

toggle this option *on* and *off* at any time while you are trying to solve the puzzle by hand.

4. *show a Knight's Tour* will find a Knight's Tour using Warnsdorff's rule and then step out the tour from the first square of any game. If a game hasn't been started, you can click on any square to show a tour after selecting this option.

5. *display a quick tour* will instantly display a Knight's Tour from your starting square if you're trying to solve the puzzle, or from any other square clicked after selecting this option.

Options 4 and 5 allow you to click on the same square repeatedly to display alternative tours from one starting square or to click on any other square to start from a different square.

What Else?

You can use the Knight's Tour to encrypt images [7]. One weakness of using the puzzle as an encryption technique is that the encoder can also be used as a decoder, so you can't employ it for the asymmetric public encryption and private decryption methods used on the Internet.

An older use of the Knight's Tour has been to hide text messages in plain sight [8]. Suppose someone wanted to encrypt the message *this message is hidden in plain sight.* First of all, the spaces and punctuation are removed so that the 31 characters in the text become the string *thismessageishiddenin-plainsight*. The message can still be read easily, but it does not have spaces that would indicate the beginning or end of a word.

You can then use a Knight's Tour, such as the one depicted in Figure 5, to hide the text. The first character of the message is placed on the square 1, the second on the square 2, and so on until the last character is placed on square 31. The remaining unused squares are then filled with random characters. Reading the hidden message one row at a time gives an indecipherable string of 64 meaningless characters, and this string can be punctuated with spaces to give the impression that a substitution cypher is being used. Random characters can also be added to the end of the message so that the hidden message doesn't always have 64 characters –

which might hint at how the text has been encrypted.

Unless someone knows how the message has been hidden, and which of the millions of possible Knight's Tours was used to hide it, it is virtually impossible to read the message. The message is padded out with redundant characters, making it difficult to decrypt by looking for anagrams, and finding the right combination of characters by chance is virtually impossible because there are ($64! / (64-31)!$) combinations of 31 characters from 64, and this is three orders of magnitude more than the number of atoms on Earth. The downside of this encryption technique is that both the encoder and the decoder need to use the same code book or program to encrypt and decrypt the message, making it much more vulnerable than modern public key/private key encryption. ∎∎∎

Info

[1] The history of the Knight's Tour: *https://www.cma.fct.unl.pt/sites/www.cma.fct.unl.pt/files/knights_tours_final.pdf*

[2] The Knight's Tour in Wikipedia: *https://en.wikipedia.org/wiki/Knight's_tour*

[3] Monte Carlo testing in Wikipedia: *https://en.wikipedia.org/wiki/Monte_Carlo_method*

[4] Dally, S. (editor). *The Century/Acorn User Book of Computer Puzzles*; Century Publications, 1984: *https://openlibrary.org/books/OL15157004M/Book_of_computer_puzzles?v=17*

[5] Code for this article (`knights_tour.py`): *ftp://ftp.linux-magazine.com/pub/listings/magazine/211*

[6] Pygame: *https://www.pygame.org/*

[7] Singha, M., A. Kakkarb, and M. Singh. "Image Encryption Scheme Based on Knight's Tour Problem." *Procedia Computer Science 70* (2015), pg. 245 – 250: *https://ac.els-cdn.com/S1877050915032457/1-s2.0-S1877050915032457-main.pdf?_tid=2f9f40be-12f8-43a3-b007-a0b80b271201&acdnat=1520931505_5f2fdc94f5e805770654f2e794c2eaf6*

[8] Tomatis, M. and M. Williamson. "Detailed analysis of the Great Parchment of Rennes-le-Château." *Indagini Su Rennes-Le-Chateau* (2008): *http://www.renneslechateau.it/index.php?sezione=studi&id=greatparchment*
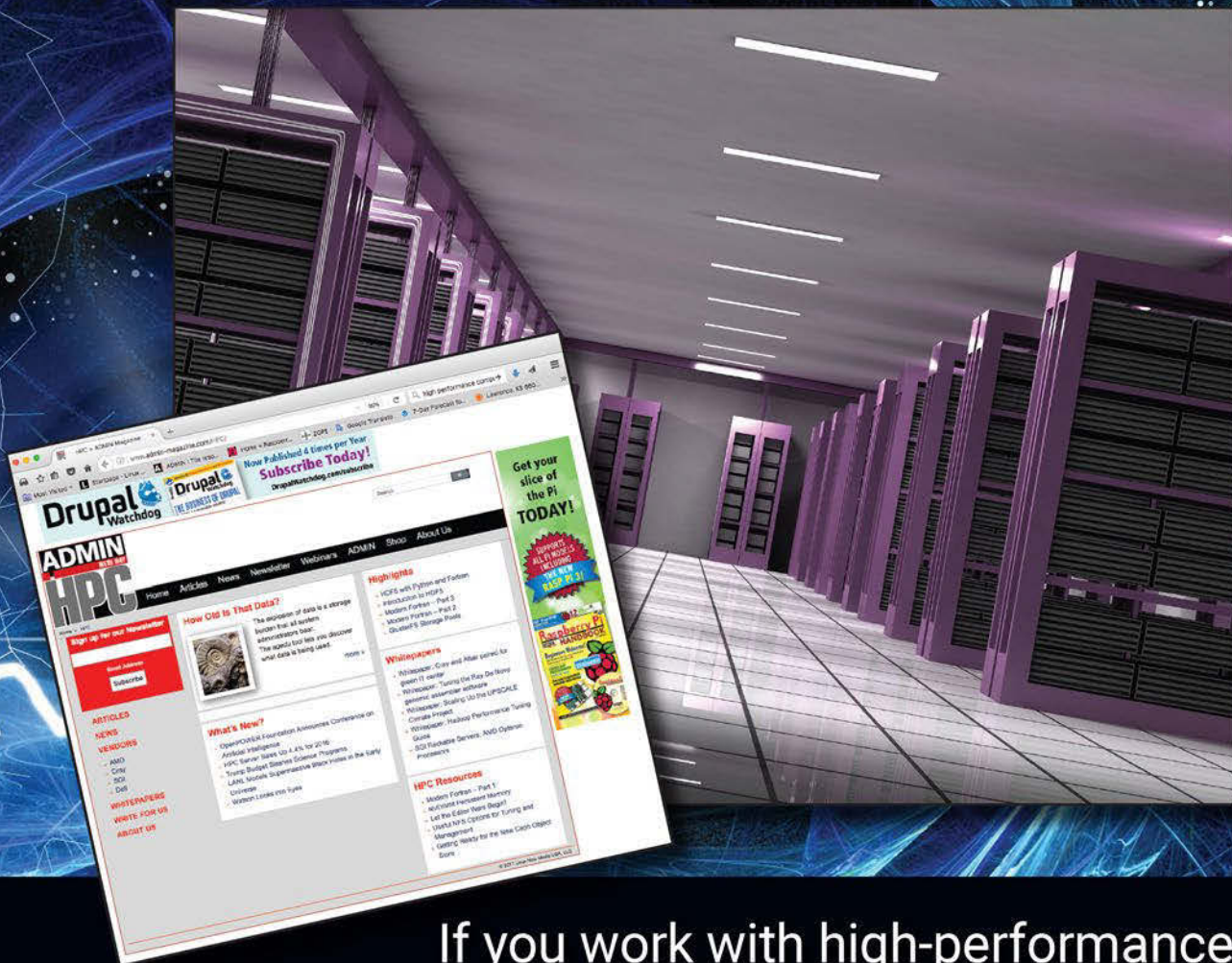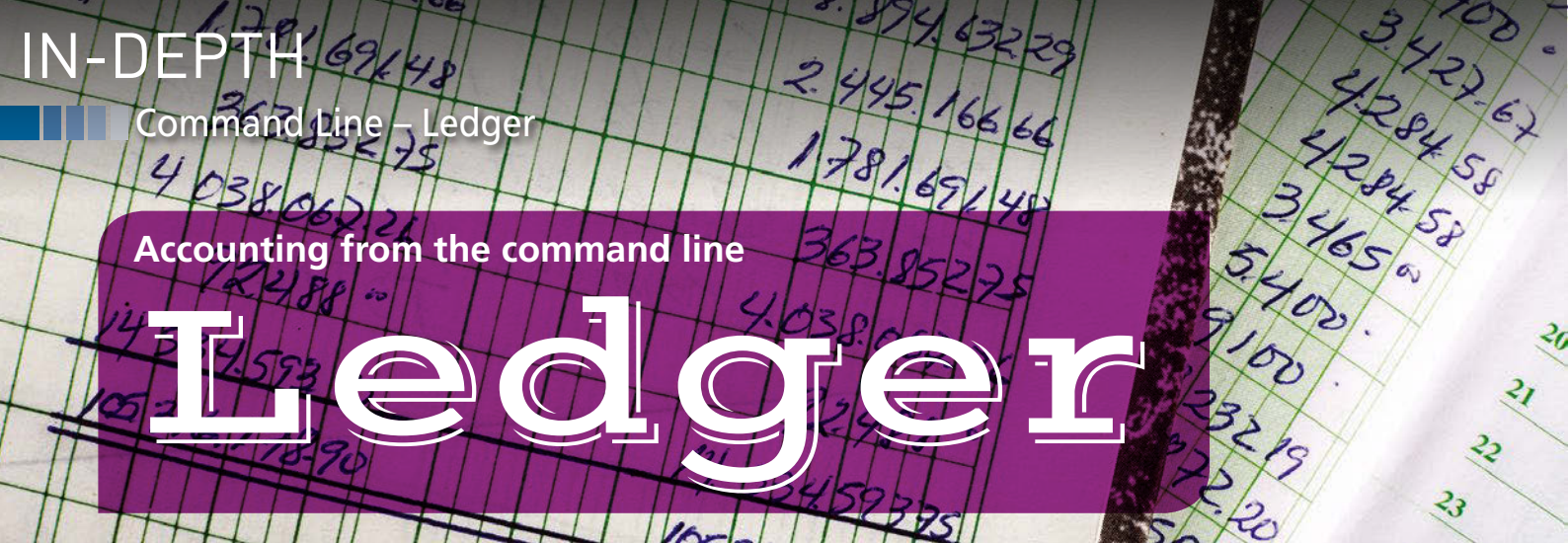
Accounting from the command line

Ledger

By learning a few basic functions, you can take your accounting to the command line.

By Bruce Byfield

A t first, an accounting program run from the command line may seem a needless complication. Surely, a desktop application like GnuCash [1], in which information is available at a glance, is easier to understand – especially if you are one of those whose brain freezes at the thought of bookkeeping? Yet perhaps the fact that Ledger [2] shows only a limited amount of information at a time is actually a point in its favor. Because it runs from a command line, Ledger shows only the information requested. Probably, it helps, too, that while Ledger functions like a database, it is a stripped down one, storing and writing information in plain text files, and generally just the essentials. If you need extras, you can write shell scripts to extend functionality. As the project's website suggests, Ledger offers a "fat-free" version of accounting. And, if you need additional information – including an introduction to basic accounting – Ledger's documentation is concise, clear, and complete.

Nor does this bare essentials approach result in oversimplification. Ledger is complete enough that a large non-profit like the Software Freedom Conservancy has used it for years to manage its financial affairs. Over the 15

years of its existence, Ledger has developed a thorough, no-nonsense approach that makes it unique among accounting software.

In fact, Ledger is such a mature application that it has dozens of options – far too many to mention here. Mercifully, though, users can generally get started with only a few options, or even none at all. To start using Ledger, users need to understand only two things: how to set up a journal, and how to generate reports to evaluate a journal. Almost everything else is a simple elaboration on these two tasks.

Creating a Journal

Ledger uses the term journal for what is usually called a ledger. A journal in Ledger is a record of incoming income and where it is assigned, such as groceries, and mortgage payments. It is called double-entry accounting, because each sum is entered twice, once as an asset, and once as an expense.

Unlike most applications, Ledger does not use a custom format. Instead, a journal is a plain text file, and postings can be made with any editor, either from the command line or the desktop. Because the file is identified by its first line:

```
; -*- ledger -*-
```

it can have any extension you prefer. All that matters is that it has information arranged in a format that Ledger can read. A sample journal [3] showing a variety of postings can be copied and pasted from the project site (Figure 1).

A journal consists of postings that lists money in and out. Listings are called accounts and are usually one of five kinds:
- Expenses: where money goes.
- Assets: where money is stored.
- Income: where money comes from.
- Liabilities: money owed.
- Equity: the real value of property.

In daily accounting, Expenses should ideally be less than or equal to Assets and Income.

Entries in a journal require a standard format, as shown in the bottom portion of Figure 1.

Figure 1: Journal entries require a standard format.

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

Lead Image © Olaf Speier, 123RF.com

```
bb@nanday:~$ ledger -f drewr3.txt balance --strict
Warning: "/home/bb/drewr3.txt", line 4: Unknown account 'Liabilities:Tithe'
Warning: "/home/bb/drewr3.txt", line 28: Unknown account 'Liabilities:Mortgage:Princip
al'
Warning: "/home/bb/drewr3.txt", line 29: Unknown account 'Expenses:Interest:Mortgage'
Warning: "/home/bb/drewr3.txt", line 30: Unknown account 'Expenses:Escrow'
Warning: "/home/bb/drewr3.txt", line 39: Unknown account 'Income:Salary'
Warning: "/home/bb/drewr3.txt", line 39: Unknown account 'Liabilities:Tithe'
Warning: "/home/bb/drewr3.txt", line 43: Unknown account 'Assets:Savings'
```

Figure 2: The `--strict` option gives you warnings about the postings that need to be examined in a journal.

Most postings begin with the date in YYYYMMDD format, followed on the same line by a brief comment that describes the posting , such as Grocery Store. On the next lines are the accounts – that is, liabilities and expenses, entered as a positive amount, and assets entered as a negative amount (since the asset is being spent). Each account must be indented by at least one space and be separated by at least two spaces or a tab from the amount. In the example in Figure 1, the amount is followed by the date on which the expense was made. Comment lines generally begin with a semicolon (;), although a number sign (#), vertical bar (|), asterisk (*), or percent sign (%) can also be used.

Ledger handles multiple currencies by dual postings. For example, if a journal is kept in US dollars, you would need a posting for buying euros, like the one shown below.

```
2017/09/23 Cash in Munich
    Assets:Cash          €50.00
    Assets:Checking      $-66.00
```

Commodities, buying and selling stocks, and fluctuating prices can also be recorded, although with slightly different details, as described in the Ledger documentation [4]. However, no matter the journal's content, consistent standard formatting is required to use it with Ledger.

Additionally, expenses and liabilities paid out should equal assets and income. You can avoid inconsistencies in accounting by defining accounts and payees at the top of the journal with postings:

```
account Expenses
account Expenses:Utilities
```

Then, if you used the `--strict` option when running `ledger balance` (Figure 2), you will receive a warning of any line in the journal in which undefined accounts are mentioned.

Using Ledger Reports

Ledger uses an unusual command structure. Instead of placing sub-commands and options immediately after the basic command, `ledger` places them after the path to the journal:

```
bb@nanday:~$ ledger -f drewr3.txt accounts
Assets:Checking
Assets:Checking:Business
Assets:Savings
Equity:Opening Balances
Expenses:Auto
Expenses:Books
Expenses:Escrow
Expenses:Food:Groceries
Expenses:Interest:Mortgage
Income:Salary
Income:Sales
Liabilities:MasterCard
Liabilities:Mortgage:Principal
Liabilities:Tithe
bb@nanday:~$ █
```

Figure 3: The Accounts report lists postings in a journal.

```
bb@nanday:~$ ledger -f drewr3.txt balance
         $ -3,804.00   Assets
         $ 1,396.00      Checking
            $ 30.00        Business
         $ -5,200.00     Savings
         $ -1,000.00   Equity:Opening Balances
         $ 6,654.00    Expenses
         $ 5,500.00      Auto
            $ 20.00       Books
           $ 300.00       Escrow
           $ 334.00       Food:Groceries
           $ 500.00       Interest:Mortgage
         $ -2,030.00   Income
         $ -2,000.00     Salary
           $ -30.00      Sales
           $ -63.60    Liabilities
           $ -20.00      MasterCard
           $ 200.00      Mortgage:Principal
          $ -243.60      Tithe
--------------------
          $ -243.60
```

Figure 4: The Balance report is one of the more basic accounting functions available in Ledger.

`ledger JOURNAL SUB-COMMAND OPTIONS`

The most common reason to run the `ledger` command is to generate a report, generally at the end of the month. Although you could check a journal manu-

ally, generating a report will help you balance a journal more easily by showing relevant information and, often, giving you less data to sort through to find inconsistencies.

Reports are defined by a sub-command. The most commonly used reports are Accounts and Balance (Figures 3 and 4). Other reports include Budget, Cleared, Commodities, Equities, Payees, Prices (for Commodities), and Register (Commodity postings). All these reports can filter information by adding a string to limit the query. For example, `balance Salary` would display information for postings that are labelled Salary. Reports can filtered by options such as `--by-payee` (`-P`), `--daily` (`-D`), or `--weekly` (`-W`). Some reports also have their own unique options, as detailed in the Ledger man page, while `source QUERY` can create SQL-like queries, and `stats QUERY` can generate statistics.

All these reports are presented as standard input and must be copied and pasted into an editor or word processor to print. However, information on filtered postings defined with `Print QUERY` can be sent directly to a printer. Additionally, sub-commands can be added to prepare a file of a report for Emacs (`org`), XML (`xml`), or a CSV spreadsheet (`convert`). Using `pricemap` can also prepare a file that creates a graph in Graph-

viz, which is useful for the prices of commodities or expenses over time.

For all reports, options are available for further filtering information, such as `--add-budget`, which displays only unbudgeted postings, or `--begin DATE`, which displays only postings on or after a certain date, to which `--end DATE` can be added to create a range of postings to display . Other options alter the default spacing, such as `--columns`, which adjusts column width (presumably after seeing problems with the default spacing) or the default columns, such as `--days-of-the-week` or `--dc`, which, when used with Register or Balance reports, arranges output in debit and credit columns (Figure 5).

These are only a handful of the available sub-commands and options. Probably, you will need to experiment and consult the Ledger documentation [5] to figure which ones will give you the information you require. The best strategy is to start with the basic reports and refine them further after studying the results.

An Application To Learn From

The above is only a general orientation to Ledger. However, it should be enough to illustrate the versatility of Ledger. Spreadsheet, database, ledger – Ledger is all of these things, yet it still

manages to be usable without a graphical interface. In fact, because it runs from the command line, it generates reports much faster than most equivalent graphical tools.

Ledger does take a long time to learn thoroughly. Yet the same can be said for other accounting programs, and users can be up and running with some basic functions far more quickly than applications that are comparable in functionality. In fact, because much of the functionality is hidden at the command line, Ledger is considerably less intimidating that other accounting programs, and therefore faster to learn. If you have been avoiding accounting apps because they seem overwhelming, then, contrary to usual expectations about the command line, Ledger just might be the tool you need to learn accounting. ∎∎∎

Info

[1] GnuCash: *http://www.gnucash.org/*

[2] Ledger: *https://www.ledger-cli.org/*

[3] Sample journal: *https://www.ledger-cli.org/3.0/doc/ledger3.html#Example-Journal-File*

[4] Further journal postings: *https://www.ledger-cli.org/3.0/doc/ledger3.html#The-Most-Basic-Entry*

[5] Documentation: *https://www.ledger-cli.org/3.0/doc/ledger3.html*

```
bb@nanday:~$ ledger -f drewr3.txt balance --dc
    $ 8,830.00      $ 12,634.00      $ -3,804.00   Assets
    $ 8,530.00       $ 7,134.00      $  1,396.00     Checking
      $ 30.00                 0        $     30.00       Business
    $ 300.00        $ 5,500.00      $ -5,200.00     Savings
            0       $ 1,000.00      $ -1,000.00   Equity:Opening Balanc
    $ 6,654.00              0        $  6,654.00   Expenses
    $ 5,500.00              0        $  5,500.00     Auto
      $ 20.00               0        $     20.00     Books
    $ 300.00               0         $    300.00     Escrow
    $ 334.00               0         $    334.00     Food:Groceries
    $ 500.00               0         $    500.00     Interest:Mortgage
            0       $ 2,030.00      $ -2,030.00   Income
            0       $ 2,000.00      $ -2,000.00     Salary
            0          $ 30.00      $    -30.00     Sales
    $ 200.00          $ 263.60      $    -63.60   Liabilities
            0          $ 20.00      $    -20.00     MasterCard
    $ 200.00               0         $   200.00     Mortgage:Principal
            0         $ 243.60      $   -243.60     Tithe
--------------------------------------------------------
    $ 15,684.00     $ 15,927.60      $  -243.60
```

Figure 5: A Balance report divided into debit and credit columns is one of the most useful options.

The sys admin's daily grind: colorls

Color Cast

The first time in our lives we got to a black-and-white Linux or Unix shell, most of us probably typed *ls* first. In a mixture of nostalgia and the knowledge that life is colorful, columnist Charly Kühnast plays a colorful trump card with *colorls*. *By Charly Kühnast*

olorls [1] is written in Ruby. If you don't have this language on your system yet, install it quickly:

```
sudo apt install ruby ruby-dev ↲
    ruby-colorize
```

Then you download a character set that you really like from Nerd Fonts [2] – say, Roboto Mono Nerd Font Regular. After unpacking the ZIP file, I moved the character set to the `/usr/share/fonts/truetype/roboto/` directory on my Ubuntu desktop; users of other distributions may need to change this path.

Why do I even get this font when there are a few dozen others preinstalled? Because Nerd Font's character sets are more extensive, containing more symbols, special characters, glyphs, and emojis than usual (Figure 1). Now I select the new font in my terminal's preferences. This fulfills the preconditions, and I can proceed to install `colorls` by typing:

```
sudo gem install colorls
```

The developers know that nobody types `colorls` 50 times a day. I recommend

that you create an `lc` alias in your `~/.bashrc`:

```
alias lc='colorls'
```

If you use a light terminal background, you should always specify `--light` or, preferably, make it permanent by appending it to the `.bashrc` alias. The output then resembles that in Figure 2 – note the cute icons and bright colors. Light-shy workers can choose a variant optimized for dark terminals by specifying `--dark`.

No Blind Faith in Color

Speaking of the downside: `colorls` is a new implementation of `ls`, which does not implement all options identically and others not at all. My big favorites `-l` and `--sort=size` fortunately work. If you type `-f`, `colorls` only displays files; `-d` only displays directories. If I want to see both, I have the choice between `--sd` (directories first – note the two dashes!) and `--sf` (files first).

If you would like a brightly colored `ls`, but have problems with `colorls` because of

missing parameters, schedule a test run with `exa` [4], which doesn't offer any fancy icons but does support almost all the `ls` parameters and adds some on top. Especially with the defaults, `exa` impresses – for example, the `-h` parameter (human readable), which outputs file sizes in human-readable units instead of bytes, is implicit. ∎∎∎

Info

[1] colorls: *https://github.com/athityakumar/colorls*

[2] Download character sets: *https://nerdfonts.com/#downloads*

[3] Nerd Font Cheat Sheet: *https://nerdfonts.com/#cheat-sheet*

[4] exa: *https://the.exa.website*

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.

Figure 1: On Nerd Font's Cheat Sheet [3], you can see the extended character sets.
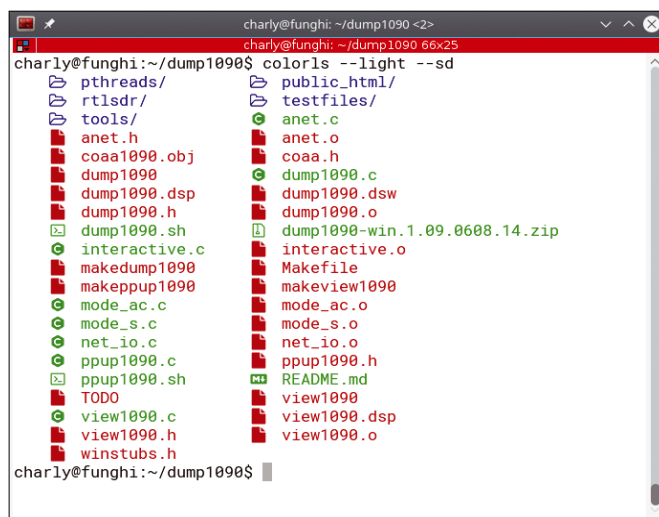
Figure 2: `colorls --light` **prints files and directories in a light color scheme.**
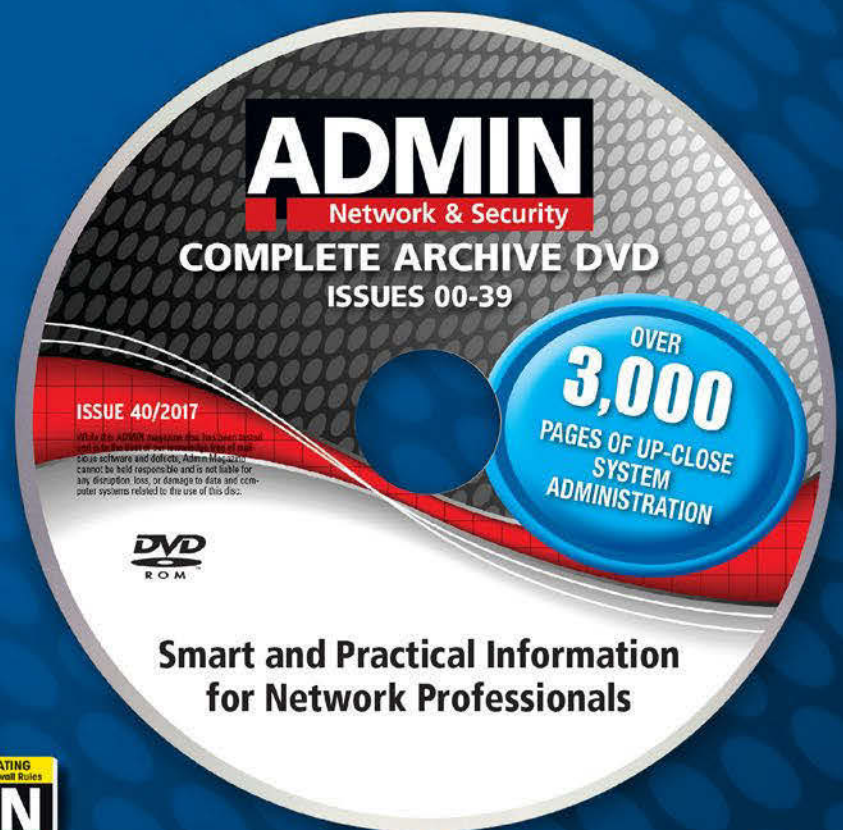
MakerSpace

Anonymous communication with PirateBox

Treasure Chest

PirateBox provides an easily accessible server for files, chats, and forum posts for meetings of all kinds. The system works without logins and protocols, so users remain anonymous.

By Christoph Langner

P irates used to have a hard time finding a suitable place for their ill-gotten booty. Today, treasures mainly exist in digital form, but finding a suitable location for a digital treasure chest is still difficult. You need to assemble and set up a small server and install and configure server services. PirateBox uses a Raspberry Pi [1] – any model will do – as a platform for your treasure trove. The system does not even require network access and works completely independently, if required.

Laying Down the Ship

The software comes in the form of a ZIP archive that you unpack and then `dd` to the memory card as you would a Raspbian image (Listing 1); or, you could use a program like Win32 Disk Imager.

On Linux, you discover the output device ID required for the `dd` command with the `lsblk` command (in Listing 1, it's `/dev/sdc/`). A 4GB SD memory card is sufficient, because the system partition does not automatically expand to

the entire disk. The system later stores data on a USB stick or a corresponding hard drive.

PirateBox does not come with a graphical interface. Therefore, you don't need to connect the Raspberry Pi used for the box to a display. Network access is not absolutely necessary, either. On a Raspberry Pi 3 (RPi3) with an integrated WiFi chip, PirateBox automatically sets up a wireless network named *PirateBox – Share Freely* (Figure 1). You need to associate with this network to configure the system.

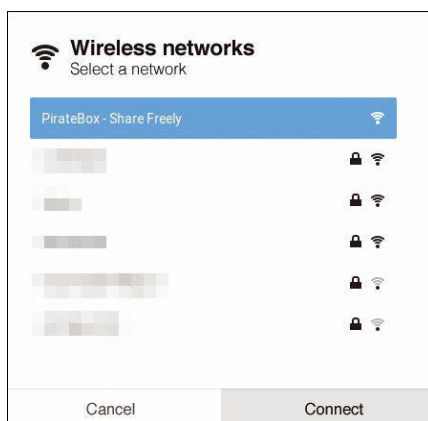Next, proceed to manage the system through SSH. To access the treasure chest, use

```
ssh alarm@piratebox.lan
```

with the `alarm` password. If you are not working in the PirateBox wireless network,

```
ssh alarm@alarmpi
```

also works. If name resolution does not work, you can discover the PirateBox's IP address on a Linux system with an ARP scanner (Listing 2). Alternatively, the classic approach is to look at the

Figure 1: After commissioning, associate with the WiFi network provided by PirateBox.

Listing 1: Installing PirateBox

```
$ sudo dd if=piratebox*.img of=/dev/sdc bs=1M; sync
2068840448 bytes (2,1 GB, 1,9 GiB) copied, 4 s, 517 MB/s
2048+0 Data sets on
2048+0 Data sets off
2147483648 bytes (2,1 GB, 2,0 GiB) copied, 4,95754 s, 433 MB/s
```

network configuration on your Internet router.

Starboard Side

After flashing the image file and logging in via SSH, PirateBox informs you of the installed version and gives instructions for your first steps (Figure 2). The first thing you should do is change the admin user password with the `passwd` command. Then, use the `date` command to enter the current time; the format is *YYYYMMDD HHMM*. This step is necessary because the Raspberry Pi does not have a battery-buffered clock and has to do without a time server in standalone operation without network access (Listing 3, lines 1-4).

As the third step, activate an image and discussion board (last line), if required. During installation, you only need to assign a password for administrative tasks.

Listing 2: Finding the IP Address

```
$ sudo arp-scan --localnet | grep Raspberry

192.168.188.25  b8:27:eb:31:86:6c    Raspberry Pi Foundation

192.168.188.26  b8:27:eb:66:ab:44    Raspberry Pi Foundation
```

Listing 3: Setting the Current Time

```
01  $ sudo timedatectl set-ntp false
02  $ sudo date -s "20180418 1841"
03  $ cd /opt/piratebox && sudo ./bin/timesave.sh .
      /conf/piratebox.conf install
04  $ sudo systemctl enable timesave
05  $ sudo /opt/piratebox/bin/board-autoconf.sh
```

Listing 4: Setting Up a Media Server

```
$ sudo cp /etc/minidlna.conf /etc/minidlna.conf.bkp

$ sudo cp /opt/piratebox/src/linux.example.minidlna.conf /etc/minidlna.conf

$ sudo systemctl start minidlna

$ sudo systemctl enable minidlna
```

You can then reach the forum from *http://piratebox.lan/board* – but only if the user is on the box's wireless LAN. The Kareha [2] software used here is strongly reminiscent of the notorious 4chan anonymous forum (Figure 3). The discussion board only offers the bare essentials, with no user administration.

For more space, connect a USB storage device, which must be formatted with a VFAT partition. Because the PirateBox system does not support the necessary Linux `mkfs.vfat` command, you have to do this on another computer first. The following command then automatically integrates the medium into the system:

```
$ sudo ⤵
  /opt/piratebox/rpi/bin/usb_share.sh
```

PirateBox offers a DLNA service (Listing 4), so multimedia files (music and videos) can be played with the appropriate players without having to provide a server or access data.

Step 6 in Figure 2 is only required if you have equipped the PirateBox Raspberry Pi with a real-time clock (RTC) in the form of a HAT add-on board.

Preparing to Board

Now fully equipped, you can deploy PirateBox at your next pirate meeting, such as a LAN party. You don't have to announce the existence of the box, because the wireless *PirateBox – Share Freely* will quickly attract the attention of the participants (see also the "On the Network" box). If a known pirate logs in, the web front end opens automatically as a hotspot page with the URL *http://piratebox.lan* (Figure 4). Thanks to its responsive design, the latest version of Pirate-Box also works on smartphones.

On the PirateBox page, participants can chat anonymously, upload data anonymously, and download data after clicking on *Files* in the page header (Figure 5). However, in our lab, I encountered limits when uploading: You have to be patient when transferring large files through the Raspberry Pi's WiFi network, and the filesystem used for the USB data carrier does not support files of more than 4GB.

You can access the media server with an appropriate DLNA client without any further configuration. On Windows, for example, foobar2000 [3] acts as a DLNA

Figure 2: PirateBox is configured via SSH. A tutorial with the first steps helps.

client (with an additional plugin). The VLC Media Player [4] runs on almost any system, and under Android, you can find free DLNA clients like Slick UPnP [5]. The combination of these two applications lets you play the videos stored on PirateBox on an Android smartphone (Figure 6).

Hoist the Mainsails

Besides the installation instructions, you will find a detailed page with further "Raspberry Pi(rate) Box Mods" in the PirateBox wiki [6]. For example, you can enable a USB WiFi stick if you do not want to use the wireless module built into the RPi3, and you will also find information on how to encrypt the WiFi network provided by PirateBox.

For security reasons, PirateBox isolates the devices logged in over WiFi from one another. In network engineering-speak, this is known as "AP isolation," with the result that data can only be exchanged between participants via PirateBox. In return, isolation provides security against unfriendly access to the client computers. If so desired, you can disable this function with a configuration in the `/opt/piratebox/conf/hostapd.conf` file (Listing 5).

If you have run out of small memory cards, and you do not want to waste space on the system SD, you can extend the system to the complete SD card by logging in to PirateBox via SSH and executing the command:

```
$ sudo ↲
   /opt/piratebox/rpi/bin/sdcard_share.sh
```

After a brief moment, the complete storage capacity of the SD card is available under `/mnt/sdshare`.

On the Network

If you do not work with PirateBox's wireless network but would rather connect the system directly to a router, the *piratebox.lan* address name resolution will not work. Instead, you can access the box with *http://alarmpi/content* and *http://alarmpi/board* or, alternatively, through the IP address. However, you must note that the upload function only works with *http://piratebox.lan*, which is hard coded in the page source. As a remedy, call the routine for uploading data directly through *http://alarmpi:8080*.
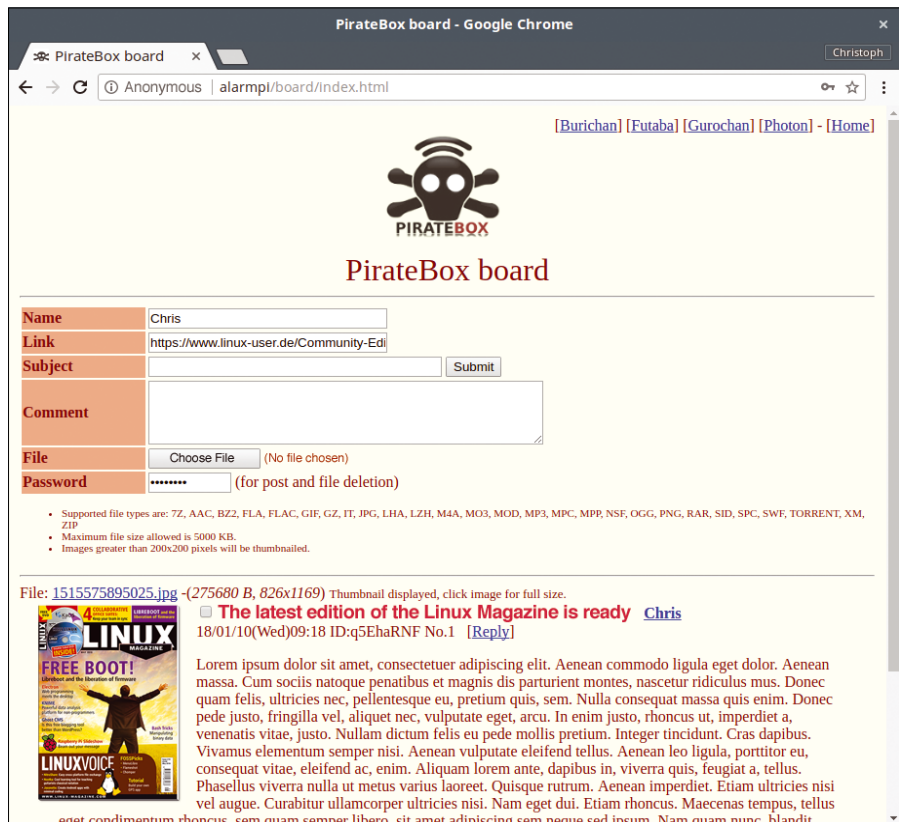
Figure 3: The Kareha forum used by PirateBox is similar to the infamous 4chan image board.
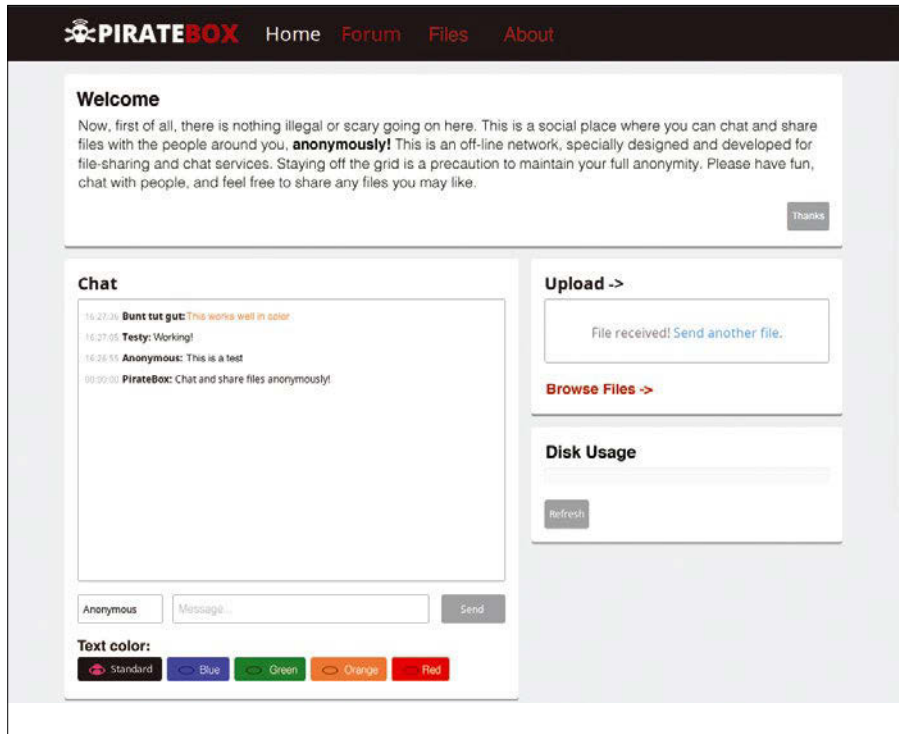
Figure 4: After logging in to the box's wireless LAN, the web front end opens automatically in a browser on compatible systems.

Listing 5: Disabling AP Isolation

```
$ sudo sed -i -e 's|ap_isolate=1|#ap_isolate=1|g' /opt/piratebox/conf/hostapd.conf
$ sudo systemctl restart piratebox
```
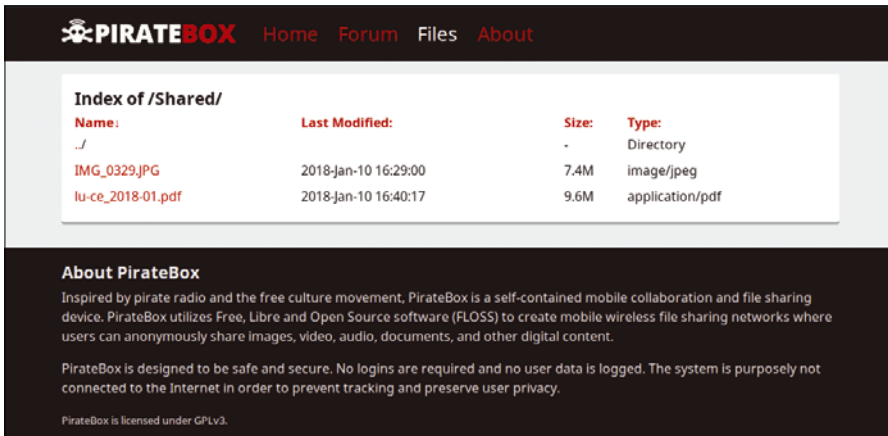
Figure 5: The uploaded data is available to all users without user administration and password protection.

Figure 6: You can stream multimedia files with Slick UPnP on an Android smartphone.

Lessons Learned

PirateBox is suitable for users who want to make an anonymous file server available to friends and acquaintances at meetings. The developers have designed the system to work without logins, and it works independently of the locally available network infrastructure: Using the Raspberry Pi's wireless LAN adapter, it sets up its own network. Additionally, PirateBox does not keep any logbooks, so it cannot trace who is chatting, has what nickname, is writing forum posts, or is loading data into the treasure chest.

Application scenarios include, for example, musicians and bands who want to offer their fans individual tracks for download at concerts or pop-up gigs. A power bank is all you need to supply the Raspberry Pi with electricity. Teachers could provide their students with a fast and easy way to use a server for projects, and presenters could use PirateBox to provide materials for a conference. ▰▰▰

Info

[1] PirateBox: *https://piratebox.cc/raspberry_pi:diy*
[2] Kareha: *http://wakaba.c3.cx/s/web/wakaba_kareha*
[3] foobar2000: *https://www.foobar2000.org*
[4] VLC Media Player: *https://www.videolan.org/vlc/*
[5] Slick UPnP: *https://play.google.com/store/apps/details?id=com.cajor.dk.dlna&hl=de*
[6] Raspberry Pi(rate)Box Mods: *https://piratebox.cc/raspberry_pi:mods*

MakerSpace

Control devices on your Rasp Pi network with text messages
Red Alert!

We show you how to control devices connected to Rasp Pi GPIO pins with text messages from an Android phone.

By Pete Metcalfe

I n places where Internet connections are not possible or too expensive, SMS text messaging can be a simple approach for monitoring and controlling your remote systems (Figure 1). Many of the mobile providers offer Internet of Things (IoT) packages for low data throughput, whereas you'd be looking at spending $1 to $5 per month for 1-5MB of data. From the hardware standpoint, many options come in either a Raspberry Pi HAT or a USB form factor (e.g., digital modem modules).

For prototyping, an Arduino phone and Node-RED [1] is a great way to jump-start your project.

Node-RED on Android

Node-RED is a graphical programming tool for IoT projects. Node-RED is installed in base Raspberry Pi images, but it can also be loaded on Linux, Windows, Mac OS, and Android systems. To load Node-RED on Android [2], you first need to load Termux [3], an Android terminal emulator app available at Google

Lead Image © Valery Kachaev, 123RF.com
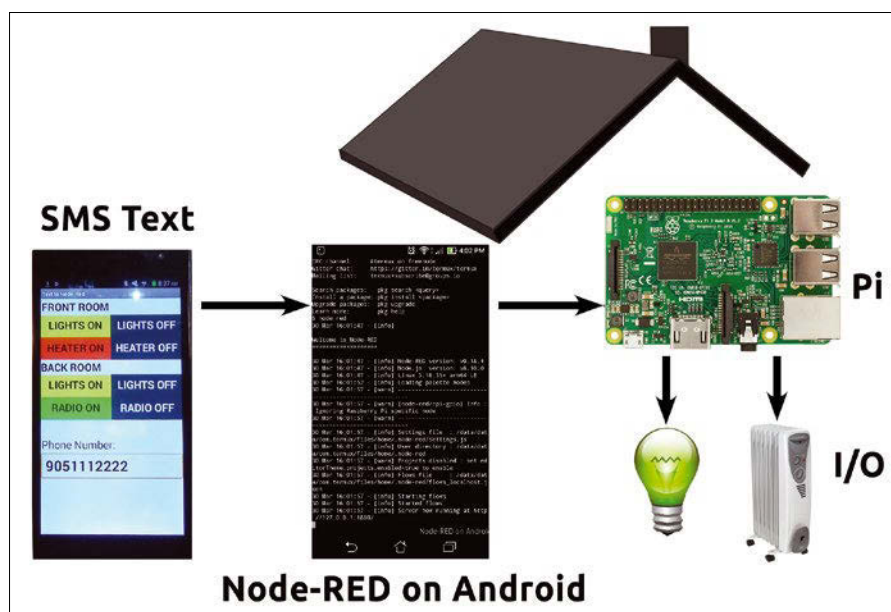
Figure 1: Control devices with SMS.

Play. After Termux is loaded, enter the following commands to install and run Node-RED:

```
apt update
apt upgrade
apt install coreutils nano nodejs
```

```
npm i -g --unsafe-perm node-red
node-red
```

Node-RED starts a graphical web interface (Figure 2) accessed by *http:// < phone_ip_address >:1880*. Extra features can be added and removed from Node-RED by selecting *Manage palette* from the hamburger menu on the right (Figure 3). For this project, I needed *terminux-api* for texting support and *bigtimer* for scheduling.

For a basic setup (Figure 4), I used:

- 1 Android phone with Termux and Node-RED
- 1 Android phone for texting
- 1 Raspberry Pi running Node-RED
- 1 NO (normally open) PowerSwitch Tail II [4] connected to a light

Scanning for Text Messages

To create a simple text message project on the Android Node-RED system, I used the following components:

- A *Big Timer* node defines how often the SMS inbox is scanned.
- An *SMS Inbox* node reads in text messages. To get the last message; select a buffer limit of *1*.
- A *Get SMS Action* function node is used with JavaScript to check for valid text messages (Listing 1).
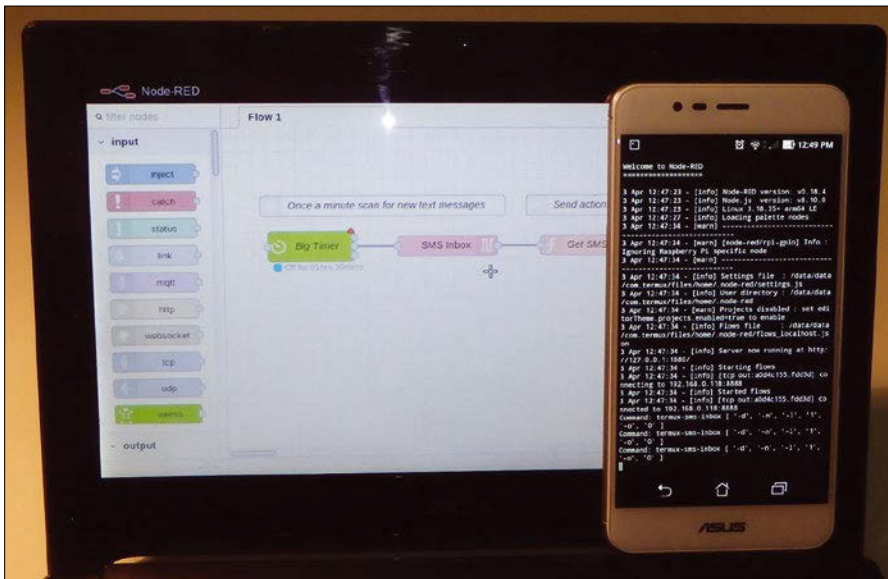
Figure 2: Android Node-RED accessed from a PC.
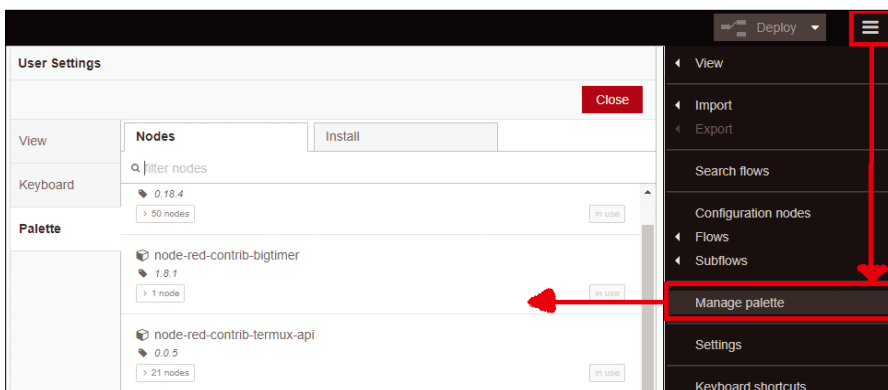
Figure 3: Installing features in Node-RED.

Listing 1: Get SMS Action Function

```
01 // look for new action texts and format the required payload
02 //
03 smslast = context.get('smslast')|| 0; //last message time variable
04
05 // Do an action if there is a new SMS message
06 if ( msg.payload[0].received != smslast) {
07     context.set('smslast', msg.payload[0].received )
08     smsbody = msg.payload[0].body;
09     // check the SMS body for valid commands
10     if (smsbody.toUpperCase().trim() == "LIGHTS ON") {
11         msg.payload = 1;  // this is passed to the Pi via TCP
12         return msg;
13     }
14     if (smsbody.toUpperCase().trim() == "LIGHTS OFF") {
15         msg.payload = 0;  // this is passed to the Pi via TCP
16         return msg;
17     }
18 }
```
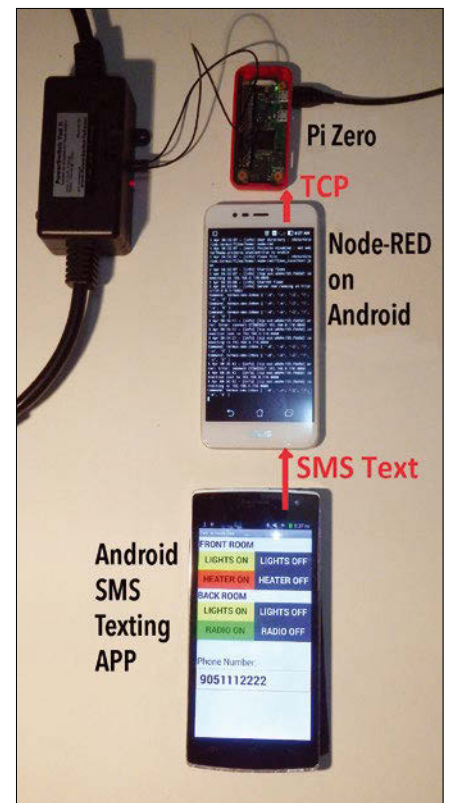

Figure 4: SMS test setup.
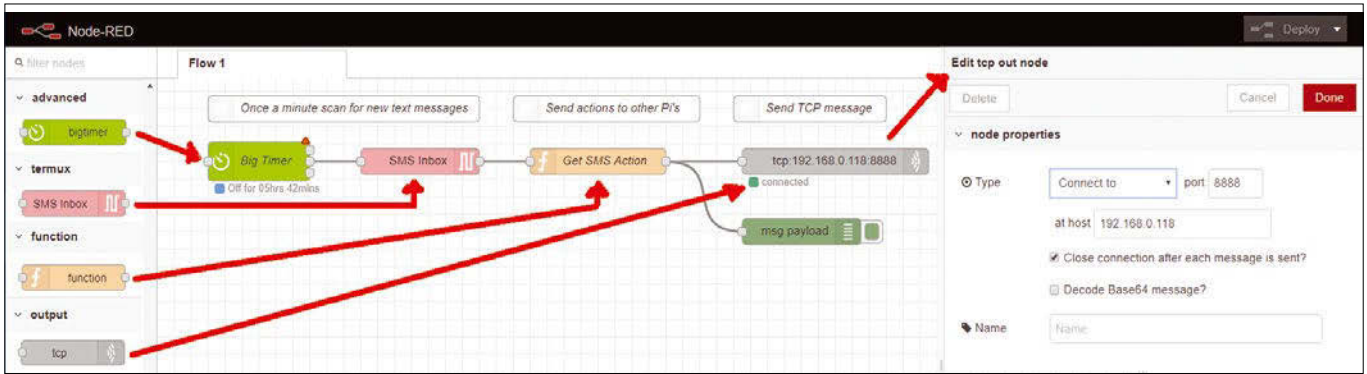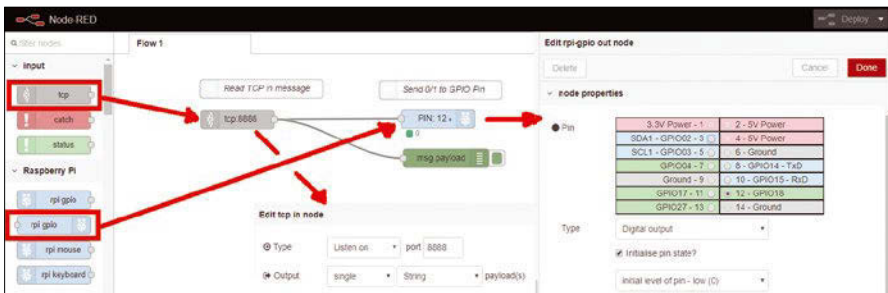
Figure 5: SMS scanning logic.

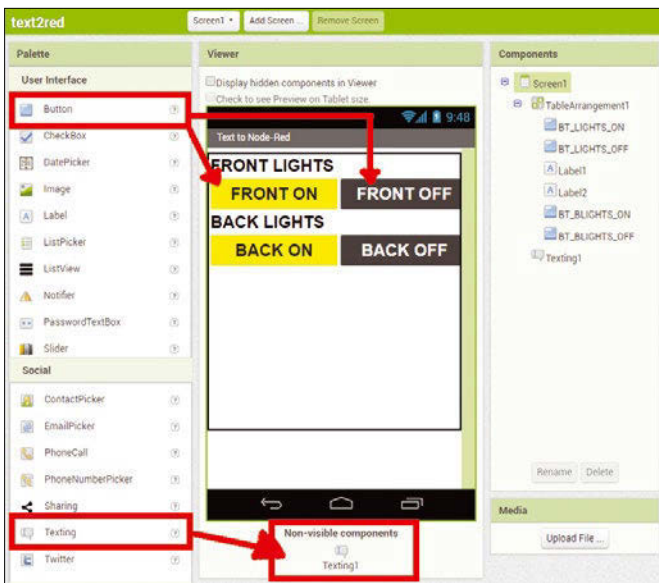**Figure 6:** Raspberry Pi TCP-to-GPIO logic.

Figure 7: App layout in MIT App Inventor.

Figure 8: Android app button logic.

cycle time of one scan per minute. The logic in Listing 1 interprets the latest text message and sends the required logic to the Raspberry Pi General Purpose I/O (GPIO) pins. The example logic uses the text message *LIGHTS ON* or *LIGHTS OFF* to control the output on GPIO pin 12.

The first time the code runs, you are prompted on the phone with *Allow Termux: API to send and view SMS messages?*. After you acknowledge this message, your Node-RED logic will run cleanly.

Reading TCP Inputs

Android phones cannot be connected directly to I/O devices, so I used TCP connections to pass commands between the Arduino phone and Raspberry Pi. Node-RED is preinstalled on most Raspberry Pi images, so no added installation is required.

On the Raspberry Pi Node-RED side (Figure 6), only a couple of nodes are required: a *tcp in* node, to read the incoming communications, and an *rpi gpio* out node, to set Raspberry Pi GPIO pins. The *tcp in* node should be configured as the same port (*8888* in this case) as the *tcp out* node. Also for this example, the output payloads (in right pane, not shown) should be set to *single* and *String*.

Texting from Android

If your text scheme is simple (e.g., turning one light on and off), you could manually just output an *ON* or *OFF* message. However if you are controlling multiple lights, heaters, and other devices, then manually typing the text messages gets awkward.

To manage multiple commands, I wrote a simple Android app in MIT

• A *tcp out* node sends a TCP message to another Node-RED system. Remember to configure the Raspberry Pi's IP address and define a port number. Figure 5 shows a basic logic setup to scan SMS periodically and pass the action by TCP. Without any configuration, the second *Big Timer* output node offers a
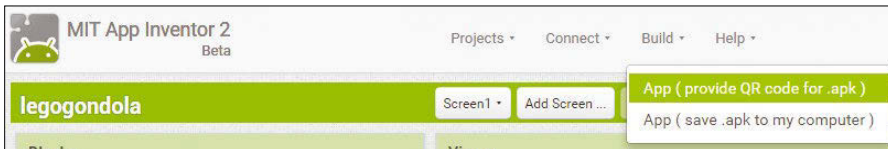
Figure 9: Building the Android app.

App Inventor [5]. App Inventor is a free web-based Android development environment that only requires a

Figure 10: Android SMS test app.

Google login for access. In about 10 minutes, I was able to get a prototype going with multiple inputs. The first step was to drag and drop some buttons from the User Interface palette onto the Viewer pane and then drag and drop a *Texting* component from the Social palette onto the view. Note that the *Texting* component will appear as a non-visible item (Figure 7).

After you have a basic layout, select the *Blocks* button on the menubar to open a logic page. Logic is created by clicking on objects in the Blocks panel and dragging and dropping the blocks onto the Viewer panel.

To have a button send a text message (Figure 8), the following pieces are required:

- The `when <Button>.Click` statement is called when the user touches the button object.
- The `set Texting1.PhoneNumber` statement defines the remote phone number.
- The `set Texting1.Message` statement defines the text message.
- The `call Texting1.SendMessage` statement sends the message.

To build the application, use the *Build* menu item (Figure 9). For my test proj-ect, I configured four devices and eight commands (Figure 10).

Final Comments

I found that Node-RED on Android is a lot faster than I expected; however, I noticed that some of the added features (e.g., Bluetooth support) only worked on the Raspberry Pi/Linux version of Node-RED.

For a final solution, I would definitely move to dedicated SMS hardware, but I found it nice to build a proof of concept test with just some basic Android phones. Also, don't forget to set Node-RED to start up automatically on a power up. ∎∎∎

Info

[1] "Fun with Node-RED" by Leah, Brook, and Pete Metcalfe. *Raspberry Pi Geek*, issue 20, 2016, pg. 90, *http://www.raspberry-pi-geek.com/Archive/2016/20/Create-amazing-Pi-apps-without-writing-code/*

[2] Node-RED on Android: *https://nodered.org/docs/platforms/android*

[3] Termux: *https://termux.com/*

[4] PowerSwitch Tail II: *http://www.powerswitchtail.com*

[5] MIT App Inventor: *http://ai2.appinventor.mit.edu*

Author

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

MakerSpace

An introduction to hacking an open keyboard
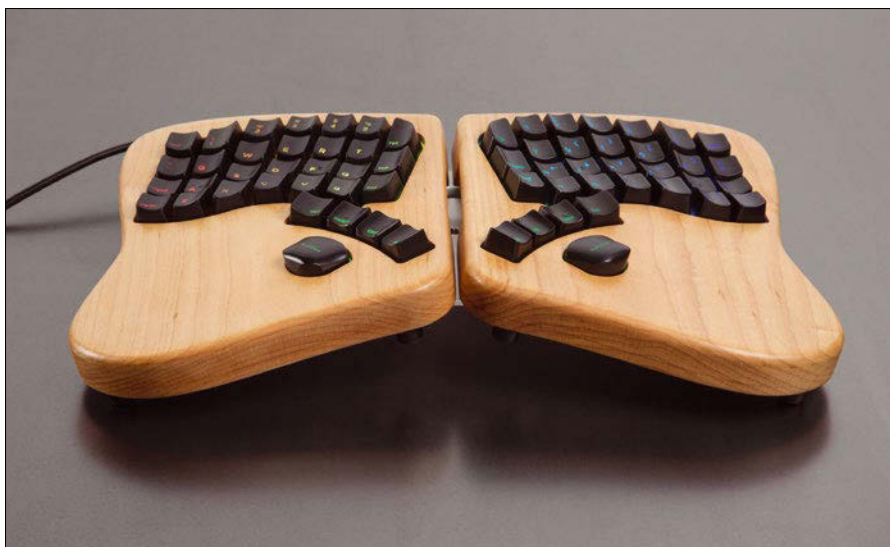
Keyboard Remodel

Reprogramming the Keyboardio Model 01's firmware is limited only by its memory and your imagination.

By Bruce Byfield

L ast month, I described the numerous features of Keyboardio's Model 01 (Figure 1) [1]. The keyboard's ergonomics, customizations, and aesthetics are joy enough for many users, but these features are just the most obvious. The Model 01 is open hardware, which means that you can program everything in the firmware to your liking, rearranging the keymap and the LED backlights and adding plugins; you are limited only by the memory of the keyboard's two ATmega32U4 Arduino microcontrollers. In the process, you can learn some of the basics of programming Arduino C for use with other devices [2], as well as some of the logic behind other open source keyboards that use some of the same code, such as those offered by Input Club.

Before you can reprogram the firmware, you need to set up the Arduino IDE and the Model 01 firmware [3]. Begin by downloading, uncompressing, and installing the Arduino IDE, following the instructions on the Keyboardio wiki. These instructions are lengthy, but each step is simple in itself. Note, though, that there are some special instructions for some distributions. In particular, the ones for Ubuntu also apply to Debian, and, I'm guessing, most Debian-derivatives as well.

Figure 1: The Model 01 keyboard.

Lead Image © 3355m, 123RF.com

Next, follow the instructions to download and install the source code for the Model 01 firmware [4]. The procedure places a copy of the firmware in `$HOME/ Arduino/Model01-Firmware` called `Model01-Firmware.ino` – a sketch, in Arduino jargon. This file can be edited in the Arduino IDE or any other text editor. When you are finished editing, the file must be flashed on to the keyboard, just as a firmware update from Keyboardio would be. A Keyboardio firmware update would overwrite your edited file, so either change its name or back up a copy of your changes as a reference. If you run into difficulties, you can restore the default firmware and try your customizations again.

To have complete freedom to customize the firmware, you need to study Arduino C. However, Keyboardio heavily comments its source code and general instructions, and numerous macros and plugins are already available. As a result, you can become moderately adept simply by reading the code already available for the Model 01 or for third-party macros and plugins. If you start with modifying existing code, you can slowly learn to write your own by looking for specific examples in existing code. The fact that the Model 01 is a specialized case makes learning easier than it might otherwise be.

Altering Keymaps

Because the Model 01 is an ergonomic keyboard, most users take a few days to adjust to typing on it. This time can be frustrating as your typing speed slows, but it can also indicate some of the first customizations to try. For example, if you find that you often overreach the *Z* key and hit the Page Down key, catapulting you from your current position in an editor, you might start by repositioning the Page Down key, or else using the key for something less distracting.

When you change the position of a key or a command, you are changing a keymap. A keymap is exactly what it sounds like: a description of what happens when you strike each key. In Keyboardio's case, the map is divided into two halves, one for each part of the keyboard. For example, Figure 2 shows the keymap for the default QWERTY arrangement of keys on the Model 01.

To change a particular key's bindings, give it a different keycode. The Model 01

```
enum { QWERTY, NUMPAD, FUNCTION }; // layers

/* This comment temporarily turns off astyle's indent enforcement
 *   so we can make the keymaps actually resemble the physical key layout better
 */
// *INDENT-OFF*

const Key keymaps[][ROWS][COLS] PROGMEM = {

  [QWERTY] = KEYMAP_STACKED
  (___,          Key_1, Key_2, Key_3, Key_4, Key_5, Key_LEDEffectNext,
   Key_Backtick, Key_Q, Key_W, Key_E, Key_R, Key_T, Key_Tab,
   Key_PageUp,   Key_A, Key_S, Key_D, Key_F, Key_G,
   Key_PageDown, Key_Z, Key_X, Key_C, Key_V, Key_B, Key_Escape,
   Key_LeftControl, Key_Backspace, Key_LeftGui, Key_LeftShift,
   ShiftToLayer(FUNCTION),

   M(MACRO_ANY),  Key_6, Key_7, Key_8,     Key_9,       Key_0,        LockLayer(NUMPAD),
   Key_Enter,     Key_Y, Key_U, Key_I,     Key_O,       Key_P,        Key_Equals,
                  Key_H, Key_J, Key_K,     Key_L,       Key_Semicolon, Key_Quote,
   Key_RightAlt,  Key_N, Key_M, Key_Comma, Key_Period,  Key_Slash,    Key_Minus,
   Key_RightShift, Key_LeftAlt, Key_Spacebar, Key_RightControl,
   ShiftToLayer(FUNCTION)),

   [NUMPAD] =  KEYMAP_STACKED
   (___, ___, ___, ___, ___, ___, ___,
    ___, ___, ___, ___, ___, ___, ___,
    ___, ___, ___, ___, ___, ___,
    ___, ___, ___, ___, ___, ___, ___,
    ___, ___, ___, ___,
    ___,

    M(MACRO_VERSION_INFO),  ___, Key_Keypad7, Key_Keypad8,  Key_Keypad9,        Key_KeypadSubtract, ___,
    ___,                         ___, Key_Keypad4, Key_Keypad5,  Key_Keypad6,        Key_KeypadAdd,      ___,
                                 ___, Key_Keypad1, Key_Keypad2,  Key_Keypad3,        Key_Equals,         Key_Quote,
    ___,                         ___, Key_Keypad0, Key_KeypadDot, Key_KeypadMultiply, Key_KeypadDivide,  Key_Enter,
    ___, ___, ___, ___,
    ___),

   [FUNCTION] =  KEYMAP_STACKED
   (___,      Key_F1,              Key_F2,      Key_F3,      Key_F4,       Key_F5,          XXX,
    Key_Tab,  ___,                 Key_mouseUp, ___,         Key_mouseBtnR, Key_mouseWarpEnd, Key_mouseWarpNE,
    Key_Home, Key_mouseL,          Key_mouseDn, Key_mouseR,  Key_mouseBtnL, Key_mouseWarpNW,
    Key_End,  Key_PrintScreen,     Key_Insert,  ___,         Key_mouseBtnM, Key_mouseWarpSW,  Key_mouseWarpSE,
    ___, Key_Delete, ___, ___,
    ___,
```

Figure 2: The source code for the default keymap in an INO file.

supports dozens of keycodes, available in three lists [5]: common characters, command codes, and controls for other devices, such as volume. You can also add common functions, such as copy and paste, with a series of brackets. For example you can reduce Ctrl + C (copy) to a single keystroke with the code `LCTRL(Key_C)` or `RCTRL(Key_C)` (in some applications, which Ctrl key you use can make a difference, but not here). For Ctrl + Shift + C (copy in a virtual terminal), the code would be `LCTRL(Key_LeftShift(Key_C))`.

Besides swapping key positions, you might try the following custom keymappings:
- Adding a Caps Lock with the code `Key_CapsLock` if you frequently type uppercase letters.
- Assigning frequently used keyboard shortcuts in commonly used applications.
- Adding a currency character, such as the symbol for euro or yen.
- Adding a command for such purposes as a system shutdown or opening a desktop menu.
- Subtracting a seldom-used key or moving it to another layer (see below).

Some of these purposes may only save you a keystroke or two at a time, but in

some cases, the advantage can quickly add up. For instance, if you are editing constantly, a single keystroke for each copy, paste, and save is both ergonomic and efficient over an entire day's work.

The largest limitation is the number of keys you can customize without giving up some functionality or shifting it to another layer. However, the Butterfly and Any keys are good places to assign new keybindings. So is the Prog key, although you will need to go through a special procedure to flash firmware if you assign something to it (see below). You can also use several macros to make multiple uses of keys (again, see below).

Another consideration is that you cannot always switch keycaps – the part that fingers press – to match your customizations, because many keys on the Model 01 are individually sculpted and are not always interchangeable with keys in other positions. An exception is the four control keys pressed by the thumb on each half of the keyboard, which are all the same shape. Otherwise, you might chose to contact Keyboardio for some blank keycaps to help minimize confusion. A blank keycap's purpose may not be obvious, but at least it will not display a former purpose. You might

even add a label to a blank keycap, although it will likely need frequent replacements.

Working with Layers

Layers are a different keymap. The default QWERTY is used automatically. Other layers, however, are activated by pressing a Command key, just as holding down the Shift key in the default layer activates uppercase letters and a small set of symbols. If you use a software-defined keyboard layout, you may also have another layout that uses a particular key to access accents, umlauts, and other diacritical marks.

The Model 01 also ships with two other layers, besides the default QWERTY. The second is the FUNCTION layer, which is activated as long as you hold down the Fn key with your palm. It gives you access to the function and arrow keys, as well as curly and square brackets. The third is NUMPAD, which activates a number pad when the Num key is toggled. Both these layers help to reduce the number of keys on the Model 01 by assigning different functions to some keys depending on which layer is currently active. Since only one layer is active at a time, any confusion is minimal, and, in any event, the Model 01 ships with a laminated sheet showing the three default layers (Figure 3).

In theory, the Model 01 can support a total of 32 layers, although in practice, it might run out of memory and keys long

before reaching these numbers. These layers could be alternative keyboard layouts or collections of macros for an often-used program like Emacs or Krita, for a favorite game, or for anything else you can imagine. You could also use the existing FUNCTION or NUMPAD for adding numerous macros, since both make light use of the left-hand portion of the keymap.

To add a layer, copy and paste an existing keymap around the existing keymaps. Then add the layer name in uppercase letters to the list that appears before the default layer in the `*.ino` file. For example, to add a layer with Vim keybindings, the list would look like this:

```
enum { QWERTY, NUMPAD, ⤵
    FUNCTION, VIM }; // layers
```

The first layer in the list is the default. If you prefer an alternative keymap like Colemak rather than QWERTY, it must be placed first in the list. However, if someone else who uses the keyboard prefers a different keymap, you will probably want another layer for their preferred keymap as well.

Additionally, each layer requires a key that activates it from any other layer. One choice is to activate a layer only so long as a key is being pressed. If you look at the QWERTY layer, you will see the code snippet for activating a layer in this way illustrated in the mapping for the Fn key: `ShiftToLayer(FUNCTION)`. Al-

ternatively, you can designate a key to toggle a layer on until the key is pressed again. This choice is illustrated in the mapping for the Num key: `LockLayer(NUMPAD)`. It would, of course, be inconvenient to always have to hold down a key while using a number pad.

Further access can be gained by defining a key with three underscores (`___`), as the Prog key is in QWERTY. When a key with this definition is pressed, it produces the character or command in the last active layer, which can be handy for accessing something quickly. You can also use `XXX` to block a key from being used on a layer, as the NUMPAD layer blocks the LED key from being used; since the NUMPAD layer uses red backlights to indicate which keys compose the number pad, you do not want this arrangement interrupted by the cycling through default light arrangements, as the LED key does on the QWERTY layer.

Editing LED Effects

The Model 01's LEDs are as programmable as the keys themselves. The LEDs are useful in dim light, but even when you touch-type, they are detected on your vision's periphery and make for more accurate typing. Some LED modes signal an event, such as connecting the keyboard or flashing the firmware. Others are accessibility aids, such as `Kaleidoscope-LED-AlphaSquare.h`, which uses the keys around the one pressed to produce the same character using all three banks of the character keys, or `Kaleidoscope-LED-Stalker.h`, which lights up a key for several seconds after it is pressed, allowing you to confirm which key was pressed. Others treat the keyboard as a light organ of rippling effects and are simply eye candy.

When you press the LED key, you can toggle through the dozen default modes. The code for each mode is stored in

```
$HOME/Arduino/hardware/keyboardio/⤵
    avr/libraries/
```

and made available by an `#include` statement at the start of the `Model01-Firmware.ino` file.

The code for these modes provide concrete examples of how to write your own code for the Model 01. However, after you have cycled through the default modes a
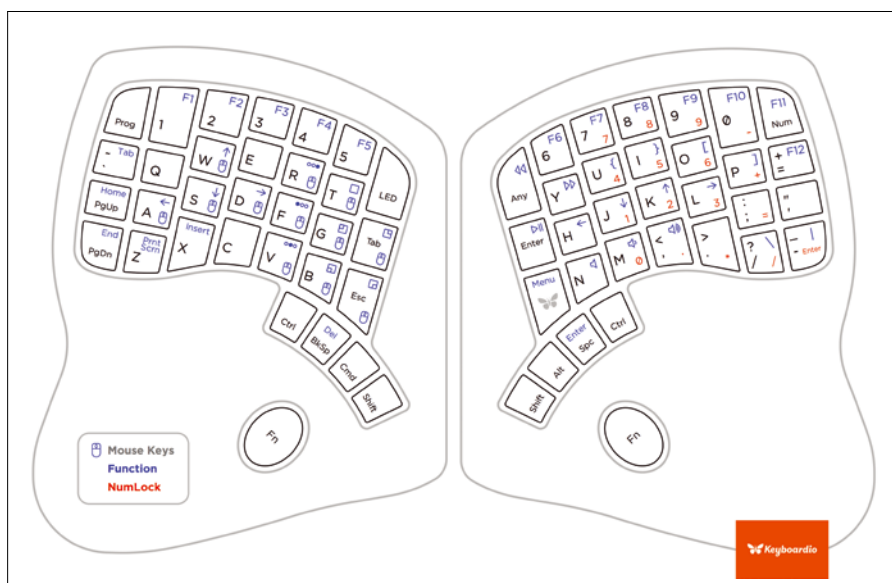
Figure 3: Keymaps for the Model 01's default layers are included with each keyboard on a laminated sheet. You can write your own layout on the other side of the sheet.

few times, your first customization of the LEDs might be to comment out the modes you no longer want, so you can toggle more quickly to the ones you actually use. If you decide on just one preferred mode, you might also search for support for an "LED off mode" and comment out `#include "LED-Off.h"` below it; then, replace the line `LEDOff.activate();` with the mode of your choice, such as `solidRed.activate();`.

Another simple customization is to edit the default solid color definitions used by other LED effects. Each value occupies a single line and includes a name for easy reference, followed by the color definition, such as:

```
static kaleidoscope::LEDSolidColor ⤶
    solidRed(160, 0, 0);
```

You can change the existing color definitions, deleting unwanted values or adding new ones to the list. You can get a color definition by opening Gimp, but remember that while Gimp uses an RGB color definition, the firmware uses a BGR definition, listing values for Blue, Green, and Red.

Macros and Plugins

Both macros and plugins are code that extend functionality. The difference is that macros are added directly to the `.ino` file, and plugins are called as libraries and stored as separate files. Often, plugins are much longer than macros, although some plugins are short. A number of third-party macros and plugins are available, and both are discussed and improved on the Keyboardio forum [6]. In installing third-party macros and plugins, you can learn the rudiments of writing your own customizations.

The exact instructions for installing either may differ in detail. For instance, some macros may need to be added before others. However, most follow the same general rules [7]. A typical example is included by default in the firmware, the Any key macro, which produces a random character and is assigned to the Any key. Using the macro requires three pieces in the firmware:

1. Add the macro code and accompanying comments after the keymaps for each layer in the `.ino` file. Follow the comments to find the proper place.

2. Search for the listing of macros and add the name of each macro, one per line so that they can be easily read; for example:

```
enum { MACRO_VERSION_INFO,
       MACRO_ANY
};
```

3. Map the name of the macro to a key; for example, `M(MACRO_ANY)`. The `M` signals that a macro is called, and the text inside the parentheses is the name of the macro.

Installing plugins requires a different general procedure [8]:

1. Add the code to:

```
$HOME/Arduino/hardware/keyboardio/⤶
    avr/libraries/
```

The simplest way to add the code is to change to this directory and then use `git clone`, specifying the URL to the code.

2. Add calls for the plugin near the top of the firmware file. For example, for `Kaleidoscope-MacrosOnTheFly`, add:

```
#include <Kaleidoscope.h>
#include <Kaleidoscope-MacrosOnTheFly.h>

void setup (){
   Kaleidoscope.use(&MacrosOnTheFly);
   Kaleidoscope.setup();
```

3. Assign the names specified in its code to a key. For example, `Kaleido-scope-MacrosOnTheFly` requires two assigned keys: `Key_MacroRec` for recording macros and `Key_MacroPlay` to playback macros. Many plugins require only one key assignment.

Two third-party plugins that are especially useful if you want to add more than a few macros are `Kaleidoscope-MacrosOnTheFly`, which not only creates macros by recording keystrokes, but also allows each key to be assigned a macro, and `Kaleidoscope-OneShot`, which turns any key into a command key. With these tools, your customizations are limited only by your imagination and the Model 01's memory.

Flashing the Firmware

You can add your changes to the keyboard in two ways. At the command line, change to the directory containing the

modified `.ino` file and enter the command `make flash`. When prompted, keep pressing the Prog key while pressing Enter. Do not release the Prog key until the process is completed. Then, unplug the keyboard and reconnect it to start the keyboard with the revised firmware. Alternatively, from the Arduino IDE, select the modified `.ino` file and click the *Upload* button while pressing the Prog key.

In both cases, do not release the Prog key until the process is completed. When the flashing is happening, the left half of the keyboard will have a red spark flash across the keys to show that the process is happening. Then, unplug the keyboard and reconnect it to start the keyboard with the revised firmware.

Should you run into trouble – for example, if you assign a command or character to the Prog key on the default layer – disconnect the keyboard and restart it while holding down the Prog key to revert to the original firmware. The left-hand circuit board has a reset button, although using it requires unscrewing the enclosure to find it. Using either one will allow you to flash another version of the firmware.

At some point, a graphical interface will be completed for simple remappings of the keys and the creation of layers, but for now, programming the Model 01 is strictly a command-line process. The bonus is that, by confronting the code head-on, you can add more complex customizations and learn how to work with other Arduino-driven devices. ∎∎∎

Info

[1] Model 01: *https://shop.keyboard.io/*

[2] Arduino C: *https://www.arduino.cc/reference/en/*

[3] Installing Arduino IDE: *https://github. com/keyboardio/Kaleidoscope/wiki/ Install-Arduino-support-on-Linux*

[4] Installing firmware: *https://github.com/ keyboardio/Model01-Firmware/blob/ master/README.md*

[5] Keycodes: *https://github.com/ keyboardio/Kaleidoscope/wiki/ Keycode-meanings*

[6] Keyboardio forum: *https://community.keyboard.io/*

[7] Installing macros: *https://github.com/ keyboardio/Kaleidoscope-Macros*

[8] Installing plugins: *https://community. keyboard.io/c/programming*

MakerSpace

Red Hat's IoT architect Peter Robinson talks about a lean new Fedora

Meet Fedora IoT

Fedora enters the IoT space with a lean new distro with atomic updates. *By Swapnil Bhartiya*

Fedora holds a special place in my digital life. It was the distribution that introduced me to the world of Linux. It is also the distribution used by Linus Torvalds himself.

Fedora is a Red Hat sponsored, community driven project that's upstream to Red Hat's commercial line of products in the Red Hat Enterprise Linux (RHEL) family. As Red Hat is diversifying the RHEL portfolio to handle different use cases, Fedora is also branching out to keep up with its commercial cousin. The Fedora distribution now has three editions – workstation, server, and cloud. The fourth edition is in the works.

During the Embedded Linux Conference & OpenIoT Summit North America 2018 (Portland), I sat down with Peter Robinson, principal IoT architect at Red Hat, who is conceptualizing the fourth member of the Fedora family.

The project has not been officially announced to the press. I was the first journalist to discuss the name of the project with Robinson, so the project is at a very early stage. Before we start seeing some code, many discussions will occur to sort out the way forward.

"A month ago, I did a presentation for the Fedora Council offering IoT as an objective for Fedora. We had a wide range of discussions about various use cases of IoT in which Fedora can play a role," said Robinson.

The Council decided it was worthwhile for Fedora to enter the IoT space. Robinson expected it to become a Fedora spin. To his surprise, the Council decided to create a new edition of Fedora with an official working group attached to it (all three existing Fedora editions have working groups attached to them). The project was approved in March. "We are now in the process of getting all the technical bits and pieces into place to make it happen," he said.

Technical Bits and Pieces

Fedora IoT is intended to run on Linux-capable IoT platforms such as the Raspberry Pi. The plan is to produce images for AArch64, ARMv7, and x86-64 systems. Robinson added, "Other architectures may come with time and interest from the community and vendors."

Robinson said he is in contact with "quite a few" hardware vendors regarding opportunities for providing support, adding, "It's early days, and I look forward to them coming and working with us in the Fedora IoT community."

The container-conscious atomic update model, with its capacity for building lean and flexible systems, is a promising technology for IoT environments, and the plan is for Fedora IoT to leverage Red Hat's investment in Project Atomic and other container-based products.

"There are still many traditional users who are used to conventional updates with `dnf` and RPMs," said Robinson. "But we are lucky because we have an entirely new canvas where we can experiment, where we can be bleeding edge without having to worry about legacy or traditional users."

Robinson has the luxury of creating a purely atomic edition, which will not have any DNA of the traditional RPM model. The cherry on top is that Fedora IoT will have the ability to layer RPMs on top of the core base of the operating system, which will be purely atomic.

Since it's a Fedora edition, it will be based on the current version of Fedora. Robinson is starting off with Fedora 28's code base and will move to 29 as it becomes stable. But Fedora has a 13 month life cycle, which is very short for IoT use cases. The Fedora Atomic Host project team has already addressed the need for a longer support model by moving the stable branch from Fedora 27 to 28 to 29, which, according to Robinson, has been a seamless roll forward. Thanks to this work, Fedora IoT will also be able to move from one major version to the next major version without any issues.

The base image of Fedora IoT is expected to have around 400-500 packages at the maximum. Additional functionalities can be added using containers and other technologies. The idea is to keep the base as small as possible. Ideally, the developers would like to keep it under 1GB or 800MB.

Current work is primarily focused on the core platform, to provide a solid foundation for others to build various IoT verticals on top. Robinson said he expects a number of open source IoT platforms and home automation frameworks will want to be part of the project.

For the Community, By the Community

Fedora IoT will be developed just like other members of the Fedora family. Although many Fedora developers happen to be Red Hat employees, the Fedora project is fully community driven.

Robinson said that the reason he is working on this project is because he wants to engage the community. "It's true that Red Hat is actively and heavily involved in the Fedora distribution, but if you look at some of the presentations from Matthew Miller, the Fedora Project Leader, more than 50% of the Fedora community comprises of non Red Hatters."

At the moment, Robinson's role at Red Hat covers a number of different areas. The work he is doing in Fedora IoT is not equivalent to a full time job. Although others from Red Hat will be involved, Robinson expects the larger open source and IoT community will want to participate in the project.

"I have a lot of ideas around IoT, but obviously there's a lot of things that I haven't thought of. People have different opinions, so I will be talking with them to work out the best route for Fedora IoT," he said. His ultimate goal is to create a small, fast, and secure edition of Fedora that's fast moving.

Does the World Need Fedora IoT

The world has no shortage of Linux-based operating systems for IoT devices, so why do we need another OS? The IoT market is emerging as a multi-billion dollar industry, so it makes sense for Red Hat to invest resources in the platform.

"There is no reason for an IoT-based Linux distribution to be that different from a traditional Linux distribution, especially when it comes to things like updates and upgrades," said Robinson. "I don't see that an IoT distribution needs to reinvent all these wheels where we've got a perfectly good solution."

The many tools and technologies of the Red Hat ecosystem could make it easier for the Fedora developers to build the new system. For instance, Robinson might get a helping hand from CoreOS, a company that Red Hat recently acquired.

CoreOS is the creator of Container Linux, a Linux distribution at the heart of the atomic updates movement.

"What really excites me about CoreOS, from an IoT point of view, is that they have a very good updates platform. There will certainly be some components in Container Linux that will be suitable for Fedora IoT," said Robinson.

When asked about comparisons with Ubuntu Core and other atomic/IoT efforts, Robinson said it's all part of the open source experience. "The IoT space is huge. Without a doubt, there are similarities, but there are differences too. I actually started talking (there are a number of videos on YouTube) about using Fedora and OSTree/Atomic for IoT before Ubuntu Core was announced. I don't follow closely what Ubuntu Core is up to, but I'm sure there's crossover in functionality. I communicate with a number of Linux distributions around a number of IoT platforms, low-level hardware layers, and open standards in order to improve the ecosystem as a whole. There are a lot of problems to be solved in the IoT space, and it's not just about competition, although a bit of healthy competition is obviously good."

According to Robinson, his work mostly involves "getting a device to boot so other people can do cool stuff on it," but with the Fedora IoT project "hopefully *I'll* get a chance to do some cool stuff as well." ◼◼◼
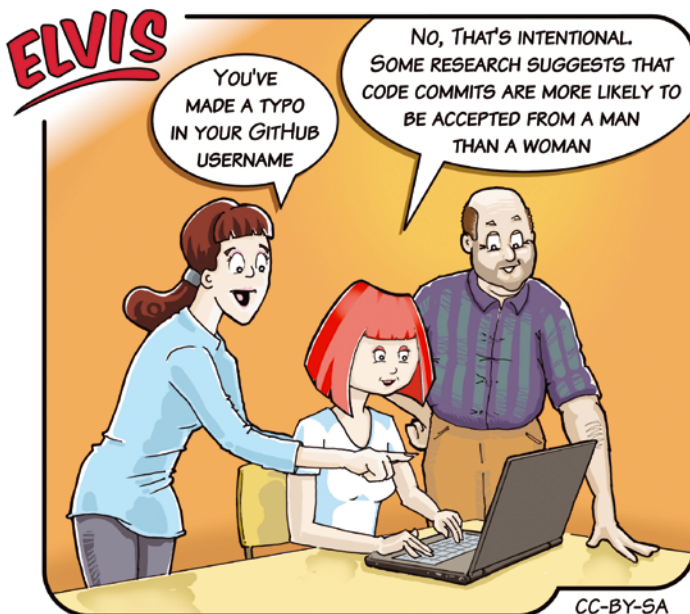
LINUXVOICE ▶

Akkk, I just broke my system! Linux gives you lots of power and with the power comes danger – a dysfunctional driver or tweak gone awry could put you on the path to an unexpected troubleshooting headache. But the power of Linux also comes with great versatility and all the energy and creativity of the FOSS community. This month we look at a tool called Resetter that was created to roll back broken Linux systems. Of course, you could always reinstall, or use a rescue DVD to un-tweak a misconfigured configuration file, but Resetter takes a different approach, letting you reset a Debian-based system to its default installation state. Also in this month's LinuxVoice, we show you how to create a podcast with Audacity and build a desktop app the easy way with QML.

Image © Olexandr Moroz, 123RF.com

MADDOG'S DOGHOUSE

Maddog gives a brief history of the last (almost) half century of *x, plus a call for spreading the love.   BY JON "MADDOG" HALL

Jon "maddog" Hall is an author, educator, computer scientist, and free software pioneer who has been a passionate advocate for Linux since 1994 when he first met Linus Torvalds and facilitated the port of Linux to a 64-bit system. He serves as president of Linux International®.

Tempus fugit

A favorite expression of mine is "Time flies when you are having fun." I usually save it for those few times when things are not going the way I hoped they would.

However, next year is 2019, and that is a very special year in my life. For in 2019, I will have been programming for 50 years … or as I like to say it, "a *half century*," which I think sounds more impressive than merely saying "50 years."

I learned to write my first program in Fortran on an IBM 1130 computer (with punched cards), through a correspondence course sponsored by the Western Electric Company, the manufacturing arm of the Bell System. I was an electrical engineering co-op student at Drexel University, and through this one course, I became "hooked" on programming.

At the same time, in a laboratory in New Jersey, two people, Ken Thompson and Dennis Ritchie, started a project that eventually became Unix. Over time, the Unix system spread throughout Bell Labs, then the rest of the world. Dennis later created a new language called C, and C became a standard language that has outlived many other languages. Eventually the philosophies and major design considerations flowed into various commercial operating systems and various flavors of BSD, GNU/Linux, xOS, iOS, and other operating systems.

In 1969, the United States Government funded a project called ARPANET to create a communications system that could withstand natural disasters and other disruptions. This network evolved into the Internet of today.

In Helsinki, Finland, little recognized at the time except by his proud parents, Linus Torvalds was born. (We will see more of him later.)

The year 1969 was very special for technology. We first put humans on the moon, and the whole world stopped for a couple of days to watch that feat, but equally important were the other things happening … we just did not recognize them at the time.

Fifteen years later, Richard Stallman (in 1984), who liked looking at the source code of computer systems, started the GNU project. In the same year, the X Window System was started at MIT.

Fast forward five more years, and the World Wide Web (WWW) was created by Tim Berners-Lee, and the Internet started moving into people's homes. Therefore, in 2019 the WWW will be 30 years old.

Another five years brought about version 1.0 of the Linux kernel, and in May 1994, I met Linus Torvalds (then 25 years old) and saw Linux for the first time, so in 2019 I will have been using GNU/Linux for a *quarter century*.

Many distributions sprang into being after version 1.0 of the kernel was released. A few distributions had started the year before, and some of those are still around, but many were released in 1994 using the first "stable" kernel. These events did not happen by magic. A lot of people thought, wrote code, learned, rethought, and rewrote code.

I am telling you all of this, almost nine months before the first day of 2019, because I want to give us all a chance to plan some really fantastic anniversary celebrations. A half century of Unix, a quarter century of GNU/Linux, and all the other triumphs (and tragedies) that happened along the way.

Take the time in 2018 to look at the last 50 years of computer science that was definitely shaped by the operating systems that started with Unix and have morphed into the mainstay of the computer industry today. Perhaps you will choose one concept of Unix or Linux and understand it really well, so that you can explain it to someone else.

In 2018, learn about one pioneer in computer science such as Charles Babbage, Ada Lovelace, Rear Admiral Grace Murray Hopper, Howard Aiken, Maurice Wilkes, or Alan Turing and what their lives were like, as well as the contributions they made.

Pick one modern-day computer scientist (please, do NOT choose ME) and learn about them. Make a presentation about their lives and contributions (PLEASE DO NOT CHOOSE ME!).

Become active in a local computer science club and help them plan a Software Freedom Day for 2019 that will be spectacular.

Read two books on how to make money with Free and Open Source Software, so in 2019 you can explain to people, "yes, there are billionaires that made their money with FOSS."

Prepare to visit one elementary school, one middle school, and one high school to talk to their computer club. Fire their imaginations! Life was not always iPhones and iPads.

Start a project that makes someone's life better using FOSS. Learn *why* you love Unix (whatever flavor) and spread the love. Let's make 2019 the year of *x.

P.S. One last thing … 1969 was the last year I ever shaved. ■■■

Clean Slate

If you have totally messed up your Debian, Ubuntu, Linux Mint, elementary OS, or Linux Deepin, you can simply reset with Resetter. BY FERDINAND THOMMES

In case of a broken system, beginners tend to grab a DVD or a USB stick and reinstall the system. This is fast, they think – and the thinking here is not wrong. But you do not have to resort to this anymore, thanks to Resetter [1]. Under the hood, Resetter is a Python script that supports resetting to the default installation for some distributions, much like the procedure for smartphones.

Resetter's way of working is quite simple: Using the appropriate manifest list, the tool determines which packages were originally installed (and in which versions) when the operating system was installed. Resetter will remove anything that does not match and reinstall the default packages according to the manifest. Besides two reset modes, the software offers another possibility to search Personal Package Archives (PPAs) directly in Launchpad and install them without any further manual work (via the *Easy Install* button).

Resetter reads the flags in Ubuntu, Linux Mint, and some other distributions to return to the state immediately after installation. The list of officially supported distributions includes the 64-bit versions of Debian 9.2 in the standard output with Gnome, Ubuntu from 14.04, Linux Mint from 17.3, elementary OS from 0.4, and Linux Deepin from 15.4.

I thoroughly tested Resetter under Linux Mint 18.3 and the current Debian "Stretch." Under Ubuntu 17.10, I could not persuade the application to cooperate; under Ubuntu MATE 17.10, even the program terminated – with bad consequences.

Easy Installation

Resetter can be set up in a flash, especially for distributions where the *gdebi* package is available. *gdebi* installs DEB packages and their dependencies downloaded from the Internet. Debian itself uses the `dpkg -i <Package>` command. However, since the command does not resolve any dependencies, you need to type `apt -f install` here to complete the installation.

In practice, you would usually download the two *add-apt-key* and *resetter* Debian packages directly from GitHub [2]. If necessary, set up *gdebi* via the package manager, and then finally use it to install the two DEBs (Listing 1). Resetter is then ready for initial use.

Before using the tool, make sure you back up, in case something goes wrong. Additionally, depending on the operating mode, Resetter may remove all user accounts and the corresponding home directories without being asked. Of course, these usually contain data that are still needed, such as browser profiles, manually customized package configurations, and the like.

Automatic or Controlled?

After starting Resetter, you will see a window with three buttons: *Easy Install*, *Automatic Reset*, and *Custom Reset* (Figure 1). I briefly described the function of the first button at the beginning; here, I am only interested in the main function of the program: resetting an installation. Resetter gives you the choice between an automatic and a user-controlled mode.

Automatic Reset mercilessly removes all user accounts and their home directories. If you select this, a warning prompt appears, reminding you of what you are doing (Figure 2). If you approve, Resetter first shows a window with all packages to be removed. Here you can exclude packages (Figure 3). If you do not do so, the tool removes all applications installed by the users, including packages in Snap format. Then, it creates a default account and adds any packages removed from the default installation again.

Custom Reset gives you more control over the cleanup: You decide which user accounts, packages, and dependencies (Figure 4) Resetter should keep. If you want to keep user accounts and their home directories, use this mode. You can enter a username and password

Figure 1: After startup, Resetter offers three options, including an automatic and a user-controlled reset.
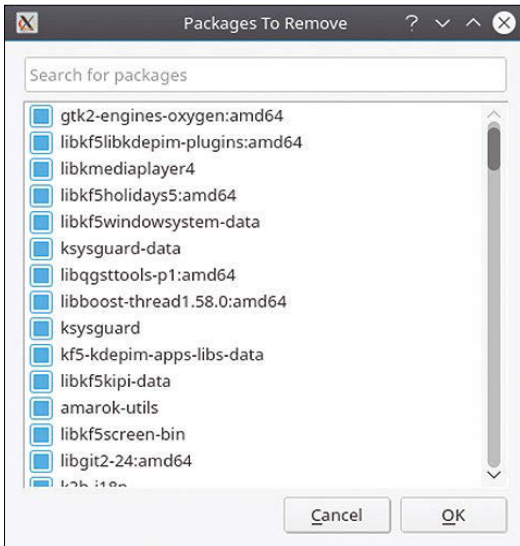

Listing 1: Installing the DEB Packages

```
$ sudo apt install gdebi
$ sudo gdebi add-apt-key_1.0-0.5_all.deb
$ sudo gdebi resetter_2.1.0-stable_all.deb
```

Figure 2: If Automatic Reset is selected, an urgent warning appears.


Figure 3: Automatic Reset offers a selection of packages to be deleted.

for the automatically created standard user. You can also instruct the tool to remove old kernels.

Slightly Inconsistent

A system in our lab with the current Debian "Stretch," with both Gnome and the KDE Plasma desktop installed, had initial start-up problems. At program launch time, Resetter output two warnings that neither a manifest nor a user list could be found; I was told that I would have to select them; otherwise, the tool would not work. After clicking on *OK*, the normal start window appeared.

If you see one or both of these warnings, you will find a remedy in the File menu at the top of the application. Clicking on *File* reveals two menu items that refer to the error messages. Select the appropriate entry for your distribution from the
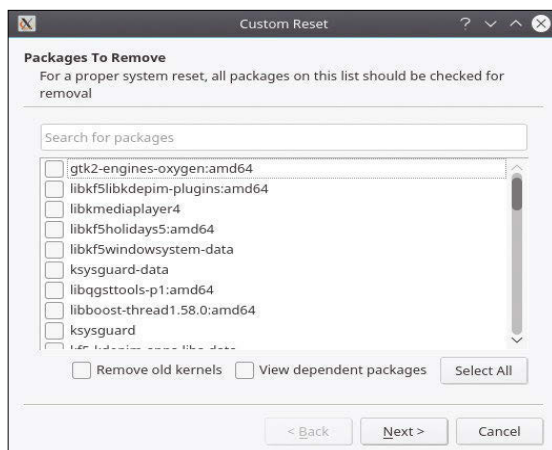
two options, then click on one of the two reset variants, and the program starts. In a window, Resetter informs you where it is in the reset process and the overall progress (Figure 5). On Debian 9, it took less than five minutes to reset.

At the end, another window appears telling you that Resetter has created a new user with the name *default* and reveals the corresponding password. The default username and password are always the same, no matter what distribution you use. You will want to create a new username with your own password immediately after a restart.

After a reboot, my Debian test system presented itself again with the standard Gnome desktop. Resetter also worked with Linux Mint 18.3 and initially with Ubuntu MATE 17.10. However, under Ubuntu MATE, the program came to a standstill at 99 percent, which led to a system that was no longer bootable to a graphical interface. Under Ubuntu 17.10, Resetter didn't even start – I only saw a blank window.

Conclusions

Resetter has the potential to save users of Debian, Ubuntu, and their derivatives the trouble of reinstalling. 2.1.0 version is still a bit unstable: Of a total of seven runs with three distributions, one failed. After our testing was finished, version 2.2.0 of Resetter was released. The new version is said to have addressed some of the bugs we encountered.

It would be great for the developers to work on further stabilizing the software as well as improving error handling. It took quite a while to connect the manifest and user list error messages (documented nowhere) with the menu items and to get started on what was ultimately quite a simple solution. ∎∎∎

Info

[1] Resetter: *https://github.com/ gaining/Resetter*

[2] Download: *https:// github.com/gaining/ Resetter/releases/ tag/v2.1.0-stable*

Figure 4: Custom Reset lets you optionally delete packages and their dependencies, as well as old kernels.
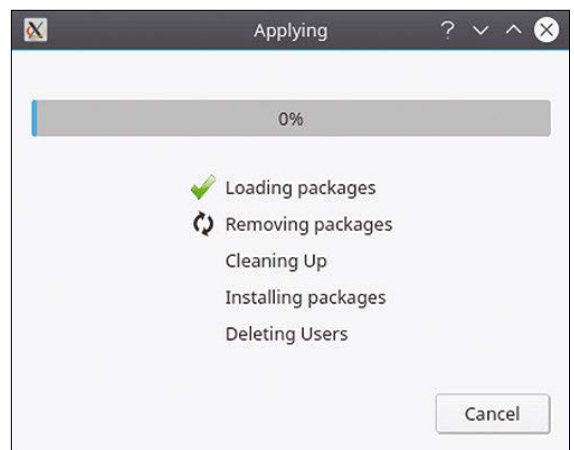
Figure 5: The software tells the user where it is in the reset process. The process only takes a few minutes.

On the Air

The glory days of podcasts might be over, but if you have something to say to your fellow humans, a digital audio recording is still a great way to deliver the message. Audacity is a free and easy tool for creating podcasts in Linux. BY MARTIN LOSCHWITZ

The term podcast was first used in 2004. At the time, the idea of publishing contributions in the form of audio files was an exciting new development in the data revolution. Since then, modern social media tools have taken some of the shine away from the podcast format. But podcasts are currently experiencing a revival thanks to streaming services such as Spotify. It doesn't matter whether you believe this 14-year-old technology is hip or not: Podcasts offer a convenient way to distribute your keynotes and thoughts in an audible form. More convenient than written text, they establish a more direct connection between the sender and receiver.

It is easy to produce a high-quality podcast. On a Linux system, you will find almost all the necessary components with the Audacity audio editing tool [1].

Audacity is aimed at (semi-)professional users and is used by many musicians. This article describes how to record and optimize podcasts with Audacity.

The Right Hardware

You already own most of the hardware you need to record podcasts. It doesn't matter whether you are using a desktop PC or a laptop. Most likely, the computer has a 3.5mm jack for connecting an external microphone. Laptops usually also come with an internal microphone that you could also use to record podcasts.

The choice of a microphone is a good place to start your podcast journey. What the microphone does not deliver in audio quality, even the best program cannot conjure up through post-processing. The microphones built into laptops are usually located near the keyboard – and are thus in close proximity to noisy components such as the CPU fan. A podcast recorded with an internal mic will almost inevitably include background noise, limiting the intelligibility and thus annoying the listeners.

If you want to record podcasts, you should consider purchasing a microphone suitable for this purpose. Even a simple headset mic, like one you might use for a video conference call, delivers sufficient audio quality for a podcast and is often available for less than $30. You do not have to confine the choice to a mic for the jack socket; a USB microphone usually offers the same quality. By the way, many external webcams also have very good microphones – your webcam might already meet your requirements.

Uncomplicated Entry

Audacity is included with all current desktop distributions and can be easily installed using the package manager. On a KDE system, Audacity installs to the K menu under the *Multimedia* entry.

Audacity 2.2 is now available, but some distributions install the less-recent Audacity 2.1. I recommend version 2.2, which comes with a revamped interface and offers many bug fixes. Ubuntu packages are available from a dedicated Ubuntu PPA [2]. PackMan [3] offers openSUSE packages.

At the top of the Audacity main window, you will find the menubar as you know it from other programs. In the bar below, you have direct access to the software's central control buttons and the tools Audacity provides for editing sound. In the third bar, you can define which devices Audacity uses for input and output. Make sure you select the right microphone.

First Recordings

Click on the icon with the red dot to start a recording. Audacity will create a new audio track in the current project. When you speak into the microphone, a waveform will appear in the audio track of the Audacity project (Figure 1). At the same time, the microphone level in the middle of the toolbar indicates what level of volume is getting through to Audacity. To pause the recording, click on the button with the white square.

The steps for recording the podcast depend on the quality requirements. In the simplest variant, just start a recording, talk into the microphone, export the resulting audio track to a common format, and publish the resulting file on the Internet.

However, if you use Audacity only as an audio recorder, you are neglecting many of its advanced features. For example, very few people are able to get through a long recording without glitches such as coughing, background noises, delay sounds (such as the classic "Uh?"), or slip-ups. Audacity offers various tools to eliminate these problems.

The most common task when editing with Audacity is to remove background noise and slip-ups. First find the position by clicking on the audio track at the approximate location and then clicking the green arrow in the upper left corner to start playback. The position of the cursor that was previously set with a click now starts moving through the audio track.

Once you have identified the location of the problem, click on the very beginning of the section you want to remove. Hold down the left mouse button and drag the mouse cursor to the right. This step creates a brightly highlighted area – you want it to extend the area to the end of the glitch. Press Delete to delete the selected item from the track.

Tip: If you notice during the recording that you have made a mistake, just pause the recording for a second. The pause button makes it much easier to identify glitches in the audio track when reworking the recording.

Optimizing the Recording

When you create podcasts in Audacity, you often need to reposition a recording in order to organize multiple recordings from different audio tracks. To reposition a recording, press F5 to select the *Move* tool (alternatively, the switch with the horizontal double arrow in the toolbar). Then click on the area you wish to move and drag it to the left or right to its new position. Finally, press F1 to return to normal selection mode.

When combining recordings from different locations or rooms, you will quickly notice that the volume of the tracks is different. Audacity offers you the possibility to normalize the volume of an audio track. Select the desired area of a soundtrack and click *Effect | Normalize*. Just confirm the preferences to normalize.

To create a transition between two audio tracks, place the tracks one after the other so that they overlap slightly. Now select one or two seconds at the end of the soundtrack you want
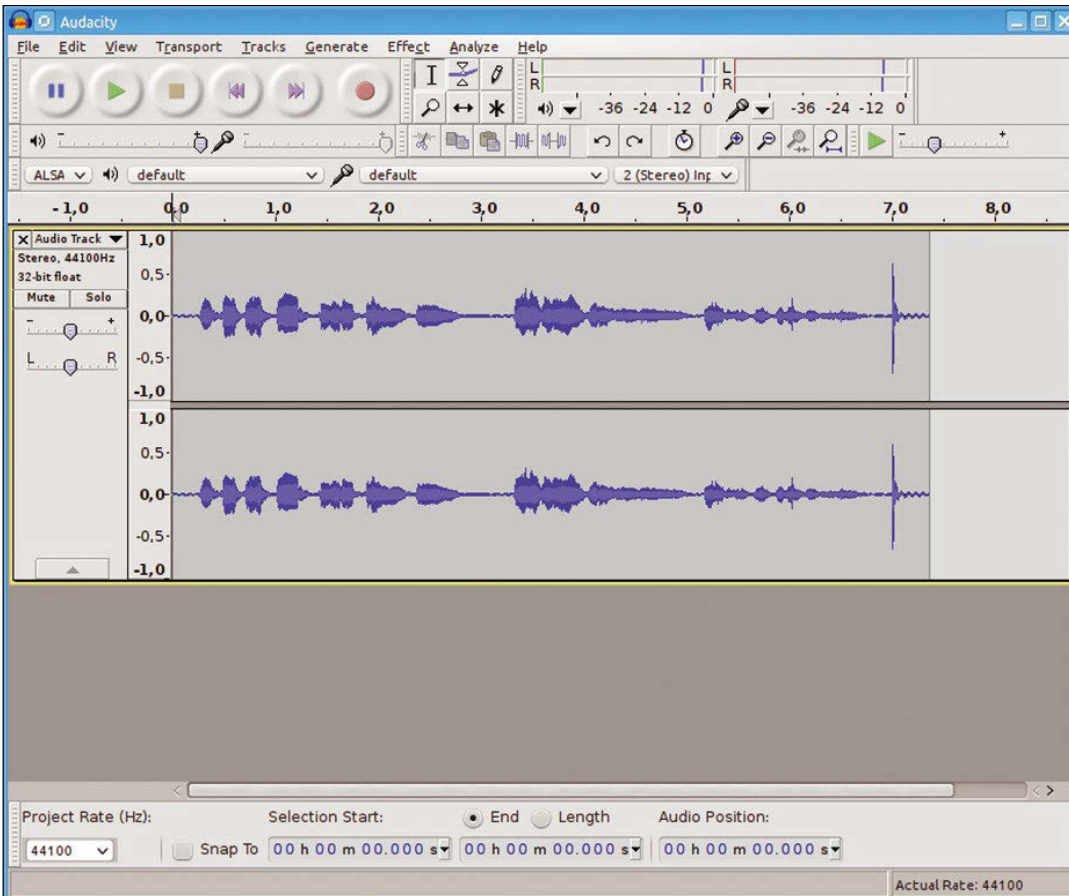
Figure 1: Audacity at work: In this example, the USB microphone connected to the computer provides a stereo track, so Audacity displays two waveforms.
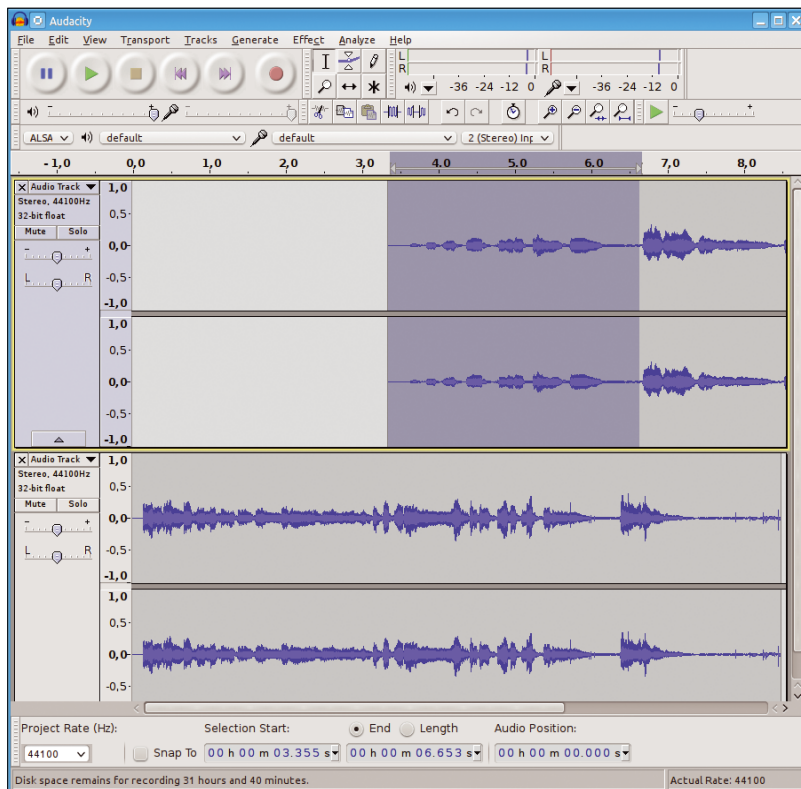
Figure 2: The same content as in Figure 1 – but with a musical intro and transitions.

Figure 3: In order to create a working podcast, you need to export to MP3 format – no problem for Audacity.

to fade out slowly and click on *Effect | Hide*. To increase the volume of the track you want to fade in, select the first one or two seconds and use the *Effect | Fade-in* option.

Intro and Outro

Experience shows that the desire for perfection grows with an increasing number of podcasts
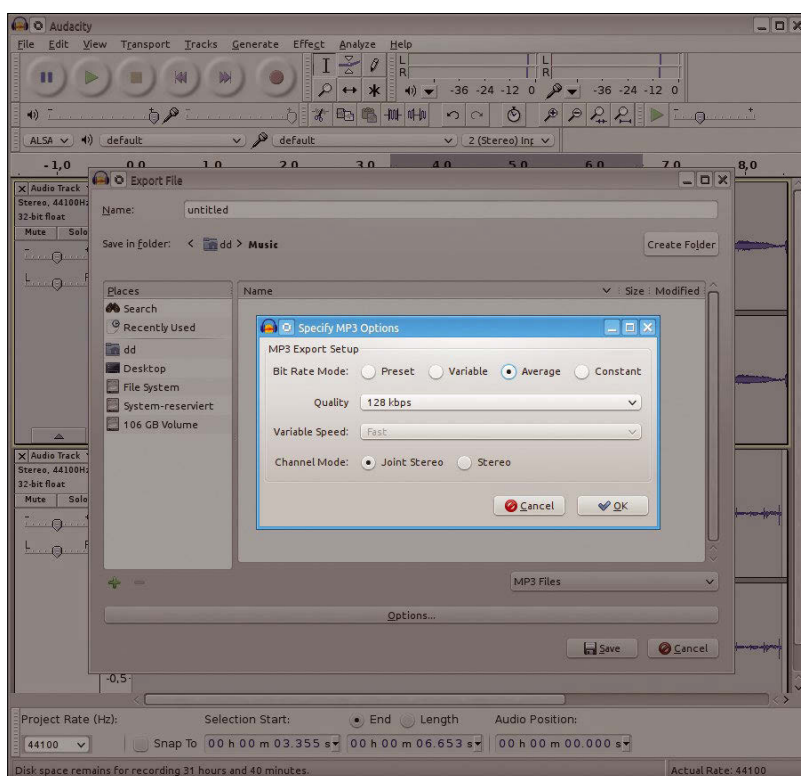
created by the producer. A common way to add a professional touch to podcasts is to add music at the beginning and end of the recordings – an intro and an outro.

No need to infringe on copyrights: A wealth of free music is available on the net, which you can use without licensing costs and which covers almost all your wishes. Just click through the Free Music Archive [4] or listen to Bensound [5] and see if some of the tracks meet your expectations.

To include the songs in the podcast, import the music files into Audacity first. To import music, download the file from the web and store the track(s) locally on your hard drive. In Audacity, click *File | Import | Audio*, navigate to the downloaded track, and confirm your selection. Shortly afterwards, the track will be available as a separate audio track in your Audacity project.

If you only want to use a small part of the work as intro or outro, edit the soundtrack exactly as you did it with your own recording. In the case of the intro, move the entire audio track to the beginning of the Audacity project. After exporting the Audacity project to a common format, the intro and outro form an integral part of your podcast (Figure 2).

Exporting Your Work

Under the hood, Audacity does not work with audio data in a common format; it uses its own raw format for saving. If you want to make your podcast available to the world, you need to export it as an MP3 or Ogg Vorbis file.

Before you export, be sure your podcast is finished: Add any intro and outro effects, remove background noises, adjust the volume throughout the podcast, and make sure no audio tracks overlap.

To convert the result to a popular audio format, click *File | Export | Export as MP3*, enter a meaningful name in the file selection window, and click *Save* or *Save as* (Figure 3). Then upload the file to the Internet, where your listeners can now enjoy it. ■■■

Info

[1] Audacity: *https://www.audacityteam.org*

[2] The Ubuntu PPA for Audacity: *https://launchpad.net/~ubuntuhandbook1/ +archive/ubuntu/audacity*

[3] Audacity 2.2 for openSUSE: *http://packman.links2linux.org/package/ audacity*

[4] Free Music Archive: *http://freemusicarchive.org*

[5] Bensound: *https://www.bensound.com*

Playful: Gravit Designer vector graphics tool

Edit vector graphics with Gravit Designer and save the results in the SVG, PNG, and JPEG formats. BY KARSTEN GÜNTHER

The World Wide Web Consortium (W3C) made a huge breakthrough with the Scalable Vector Graphics (SVG) format, which supports far more than just employing and positioning standard objects. When the committee first published the XML-based format in 2001, the commercial programs prevalent at the time ignored it and continued to rely on their own, mostly proprietary formats. Even when web browsers started supporting SVG images, it was not enough to encourage more programs to work with this format.

However, Inkscape – a fairly comprehensive graphics program that is a mixture of two-thirds drawing program and one-third editor – could generate and load SVG data. The application still provides the most comprehensive features for this format, although it does not support all aspects of SVG.

Nevertheless, the powerful, well-documented, and widely used Inkscape has long left little room for other free vector graphics tools. The Xara Group made a less successful attempt with Xara Xtreme, which found very few friends in the Linux camp. Version 0.7 of the proprietary program is still available as a 32-bit binary in various repositories.

Meantime, well removed from the mainstream, another program with quite interesting features – Gravit Designer [1] – is being developed, and it promises to be fully fledged vector graphic design software. This article is based on version (3.2.6), which was the current version at the time I wrote this article. Since then, the developers have released another major version, Gravit Designer 3.3, which includes "critical bug fixes" and "some long-requested features from the community." See the announcement at the Gravit Designer website for more on the new release [2]. Gravit Designer is not intended as a replacement for Inkscape but is more oriented toward the commercial design tool Sketch. Therefore, the main focus is on importing Sketch files, not SVG graphics.

Gravit Designer works across platforms because the developers implemented a JavaScript graphics engine, and the program itself runs in a kind of browser. You receive the proprietary application free of charge in the form of a ZIP AppImage (executable container) [3]. An AppImage is a complete filesystem that you mount on startup and contains the software and all required libraries. After unpacking the 64MB AppImage, you can place it in any directory (typically `/opt/`) and start it from there.

First Impressions

Whereas Inkscape shines through an incredibly extensive menu and many functions, Gravit Designer limits itself to relatively few functions in a clear-cut and tidy interface (Figure 1). However, the differences go much further than the surface shows.

Inkscape was developed as a classic design tool for the desktop. Gravit Designer, on the other hand, goes the way of modern apps,

Figure 1: Gravit Designer comes as a container application running in a (hidden) browser, which explains the unusual appearance.
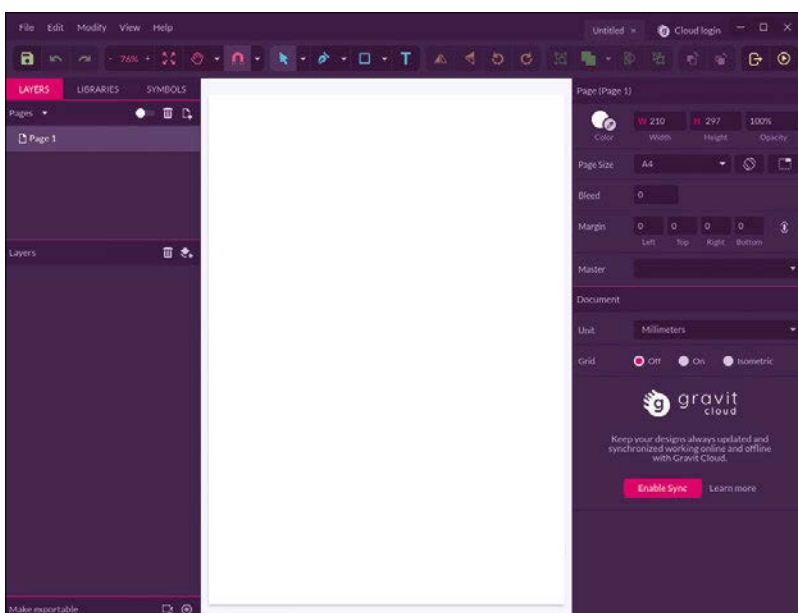
wherein the Internet provides storage and supports the exchange of information between instances. The software preferentially uses the manufacturer's own cloud; information is only available in the form of video tutorials on YouTube, along with a couple on Facebook and some other web portals [4].

You can avoid putting your data in the cloud by saving a file to disk with *File | Save to file*; however, templates and tutorials are only available online. To use the templates, you need an account for the Gravit GmbH cloud, which causes problems if you're using the Tor browser, for example.

Gravit Designer also uses its own proprietary file format to store data, although it is compatible with other formats, including SVG, with some restrictions. The software explicitly supports SVG, but not very well. When exporting, the program offers both uncompressed and compressed variants in the File menu, but the results are not convincing (Figure 2).

Practice

As with any new tool, it takes time to work effectively with Gravit Designer. As a test case, I created a small invitation to a dinner party. It contains many common elements in the form of vector graphics, including superimposed layers, curved fonts (i.e., text to path), and embedded bitmap graphics.

Implementing the software as a JavaScript applet in the browser forced the developers to design all functions and dialogs themselves, which leads to some dialogs appearing unstructured, somewhat clumsy, or rudimentary. A typical example is the color selection dialogs. Gravit Designer has only three options for defining colors, whereas Gimp (depending on the extension), for example,

Figure 2: When saving a file as SVG from Gravit Designer, important information is irretrievably lost. (Left) The original graphic created with the application. (Right) The SVG file.
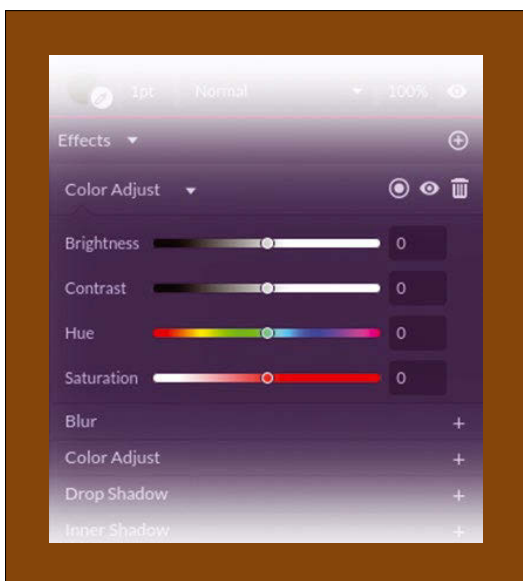
Figure 3: By default, Gravit Designer uses a variant of the HSV color model.
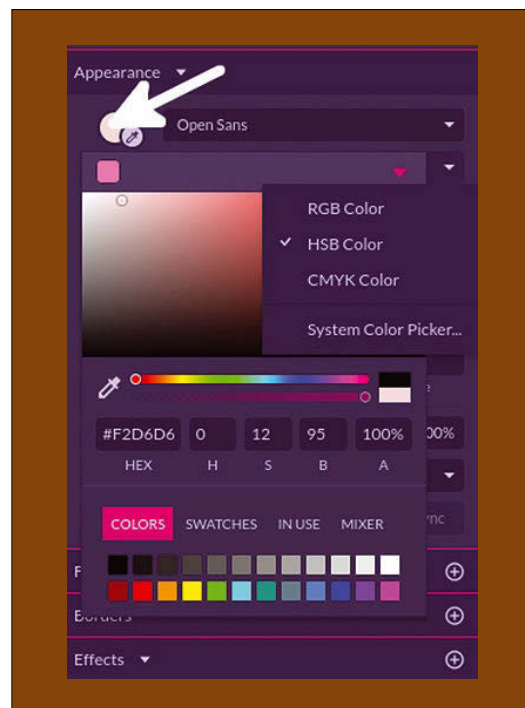
Figure 4: Additional color models are only available after clicking on the color selector (arrow).

offers five to six variants. The preset method also combines brightness and contrast in the same dialog (Figure 3).

The rarely used hue-saturation-brightness (HSB) color model, a hue-saturation-value (HSV) variant, is used; the last component is realized in the form of brightness and contrast. You only see the other color models when you click on the small color area next to the font family. This action opens a second dialog (Figure 4) with additional possibilities to set colors.

Gravit Designer creates layers automatically. Alternatively, you can create them manually, move them into the layer stack, or rename them with a mouse click. By default, layers are given the name of the tool you used to create them, which leads to the accumulation of many layers with the same name (e.g., *Path*) that cannot be distinguished from each other. To change the name of a layer, double-click the name to enable it.

Figure 5 shows the fairly simple Layers dialog, in which you can only directly influence locking and visibility. Everything else is controlled by the options of the elements in your graphic, such as layer modes or opacity. The lock and visibility icons only appear when mousing over a layer. The program shows where the layer is located in the image by highlighting the corresponding element with a box.

Because sequence and relative position play a decisive role when using objects, layers have corresponding functions in their context menus that let you group previously selected objects, which then appear as one unit.

When creating the design for the test, I used the *Text to Path* function (the name in Gimp). This often-used feature causes continuous text to wind along a previously defined path. Gravit Designer implements the function well, but in an unusual way: With a path tool, you first create a line along which you then guide the text. You no longer need the line itself.

Enable the line by double-clicking; the software then provides an enclosed area with a rectangular frame. Now select the Text tool and move the mouse cursor toward the path you drew. As soon as it turns red, click on the line where you want to start the text. The program automatically inserts *Your text here*, which you double-click to edit.

You will find the other settings for this function in the Appearance pane. *Scale font on resizing* lets you set where the font appears relative to the path. However, if something goes wrong here, *Undo* has a problem restoring the previous state. You'll find that this happens often in more complex images and dramatically increases the computing time.

This example shows that the combination of the container format and the implementation in JavaScript is not exactly optimal. Especially for situations in which RAM is in short supply, Gravit Designer's speed can be very slow, which is even more apparent in large projects.

A number of further steps follow if you also want to select the shape, adjust the size, and change the color of the font. As is customary today, Gravit Designer draws on and supports all
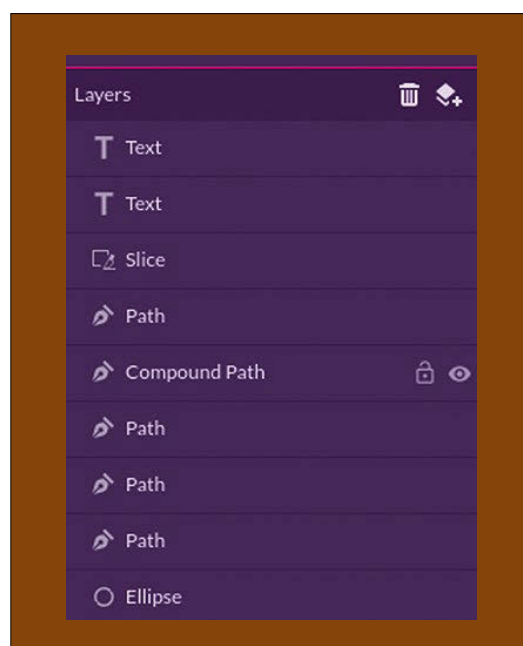
Figure 5: In Gravit Designer, layers serve more as containers for objects than as active graphic elements.
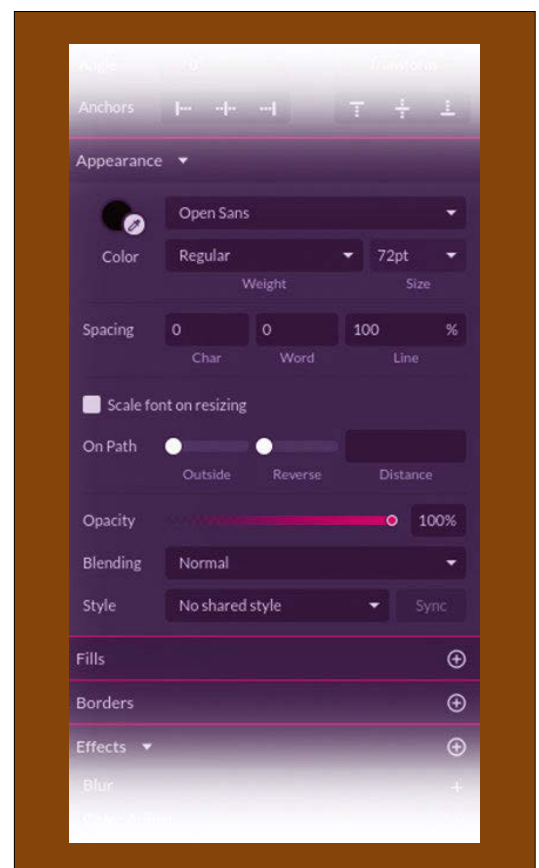
Figure 6: Once you have positioned the text, you can adjust many font details.

the fonts available on the system. In the Appearance pane, you can select the font family and its attributes (Figure 6).

The program is very stingy when it comes to font sizes, which go up to 72 point by default; other values can only be entered directly in the *Scale* field and with the mouse wheel. At least parts of the font can be selected and – within a font family – provided with different attributes, such as bold, italic, or another color.

The software supports transparency, albeit indirectly: Select the *Scale font on resizing | Appearance | Blending | Inverse Mask* layer mode. Whereas *Mask* reduces the display to the areas covered by the current layer, *Inverse Mask* makes the areas covered transparent.

In the File menu you will find two ways to integrate or link bitmap graphics under *Place Image* and *Link Image.* You can adjust the size and position of the images, but Gravit Designer does not provide for any further editing. You might need to crop the images in advance, hide the edges, and adjust the brightness and contrast.

Internal

All told, the process is very simple for smaller projects: As usual with vector graphics, you assemble the images from objects. In the toolbar, you will find a collection of standard elements under the square. Besides line, rectangle, and ellipse, you will find polygon, triangles, and stars; spirals and other common objects are missing (Figure 7). Only the first three allow you to select them directly using shortcuts; for the others, you need to use the drop-down menu.

By default, Gravit Designer fills all objects. You can control many properties with parameters in the Appearance pane on the side, such as the fill color and coverage ("Layer mode") under *Fills.* The designer solves the functions for rotation and size adjustment in an exemplary way by providing handles in the main window to rotate and scale each object directly (Figure 8).

When it comes to tools for paths – *Pen*, *Bezigon* (Bezier polygon), *Freehand*, *Shading* (closed freehand objects), and *Knife* – the situation is similar. The interesting *Knife* tool makes it possible to cut out parts of objects for further manipulation.

However, these tools are unlikely to be sufficient for more demanding work. In this context, it's interesting that each application of a tool creates a new layer, which serves as both an object and a journal (*History*), making it easy to apply subsequent changes to the images.

The software provides a *Drop Shadow* and an *Inner Shadow* for all objects. Two dialogs are avail-

Figure 7: The selection of standard objects from which you construct a vector graphic is spartan.
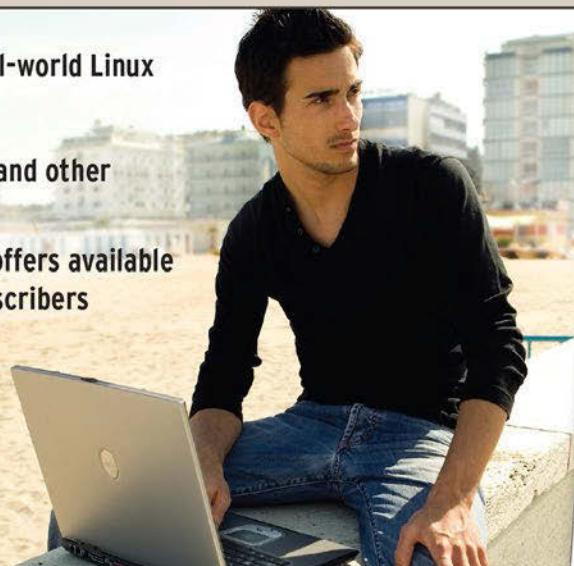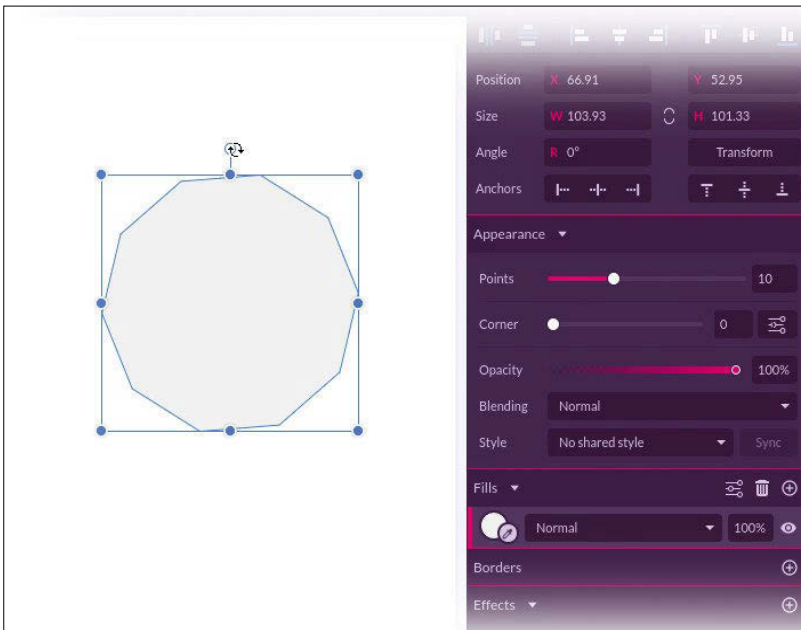
Figure 8: The JavaScript-based application provides special handles for resizing and rotating objects.

able in the sidebar. The only other possibilities are to select colors (*Color Adjustment*) or to *Blur* objects. The designer has no more direct manipulations planned. Compared with Inkscape, this is a very small toolkit.

Selection tools look a bit better. Besides the cursor (*Pointer*), which selects precisely one object, you can *Subselect*, use the lasso, and select layers or *Slices*.

If you run out of space in the main window, the *F* key hides all the controls. Pressing Alt+Enter switches the program to full-screen mode but also

places a considerable load on resources (computing time and main memory).

More Effects!

At first glance, the selection of effects (bottom right) appears very sparse. The software initially only offers the functions already mentioned when activating an object or group. A number of effects are found under *More*, which somewhat alleviates the spartan impression. However, the design of the dialog is unfortunate (Figure 9).

For example, you can't get an overview of all the functions: In a drop-down menu, you first need to select which additional effects you want to see before choosing a single effect from the selection that the program applies directly without a preview – at least the action can be undone. The visibility icon, which only appears when you mouse over the title bar of the effects dialog, is a way to disable selected effects temporarily without deleting them directly.

The window is usually too small to show all the Adjustments options, and the preview images are also too small to be useful. Only trial and error will help you here.

Conclusions

Gravit Designer is a cloud-integrated JavaScript application that generates and processes graphics in the open SVG format. It does not come close to the abilities of Inkscape, but it also requires less training.

Although it is a free program, Gravit Designer is proprietary software. If you opt to use the Gravit Cloud, you deliver yourself and your data to the manufacturer, for better or worse. Apart from tutorials in the form of videos, you will find almost no structured information on how to use the program.

The intended target group is certainly not professional designers. Even for users who only occasionally use the software, the program will probably leave you with mixed impressions. It is very annoying that the option of saving files in SVG does not work well enough to use the software in conjunction with other programs. Thus, it remains an isolated solution. ∎∎∎
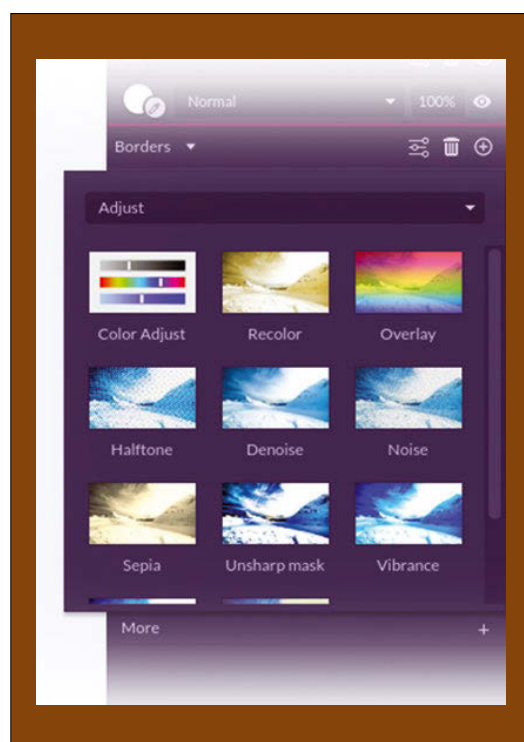
Figure 9: Gravit Designer's dialog for offering additional effects has an unfortunate design.

Info

[1]   Gravit Designer: *https://gravit.io*

[2]   Download: *https://medium.com/gravitdesigner/back-from-the-dead-gravit-designer-3-3-360b38c08127*

[3]   Download: *https://designer.io/#download*

[4]   Tutorials: *https://medium.com/gravitdesigner/gravit-designer-tips-tricks-ee6a1353aa3*

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 tool

Krita 4.0

Not since Deluxe Paint was bundled with the humble Commodore Amiga has such a wonderful drawing package been available for free. And unlike Deluxe Paint, Krita really is free in both the cost and the freedom sense, although the team would greatly appreciate any contribution you can afford. Krita's feature list obviously eclipses a package that became first available over 30 years ago, but both Deluxe Paint and Krita share the same addictive compulsion that makes even non-artists click and drag virtual paint brushes across a virtual canvas. Krita 4 is another huge step forward for a team that goes from strength to strength with each release, bundling lots of visible changes rather than just behind the scenes refactoring.

Top of these visual changes is the vastly improved brush preset editor alongside freshly reorganized brush sets. The new ed-

itor finally previews what your new brush stroke will look like, updating in real time as you play with the many parameters. I should also mention that the rendering of brush strokes is now multithreaded, vastly improving the brushes' drawing speed, especially now that there's an even larger maximum brush size. You can also assign specific brush presets to the shortcut slots, and a quick settings palette lets you quickly select specific values for common parameters, rather than having to dial them in every time. This has all been augmented with a scripting overhaul that includes a new scripting engine (thanks to a Kickstarter stretch goal!).

Your project will now also be saved transparently in the background, without even interrupting your workflow, and the revamped vector tools give Inkscape a run for its money, especially when combined with the clever fill tools that take color hints for an area and magically colorize from there. No less important is that the default format is now SVG, rather than the OpenDocument Graphics format used in version 3.x, and the quality of the vector output is much improved as a result. The GUI too has improved in many small ways. Panels are now more compact, and many elements can be scaled (such as the brush presets). Improvements can also be seen in the new vector tools and the slightly darker color theme.

We also must mention the fabulous release notes for 4.0. Not only do these step you through every change for this release, all excellently written, but each feature is often accompanied by excellent illustrations, many of which take the form of animated GIFs. When so many developers seemingly release hard-wrought new features with barely a mention in a changelog, this is truly a refreshing change. Repeat after us: "A feature doesn't exist if it isn't documented." Thanks Krita team!
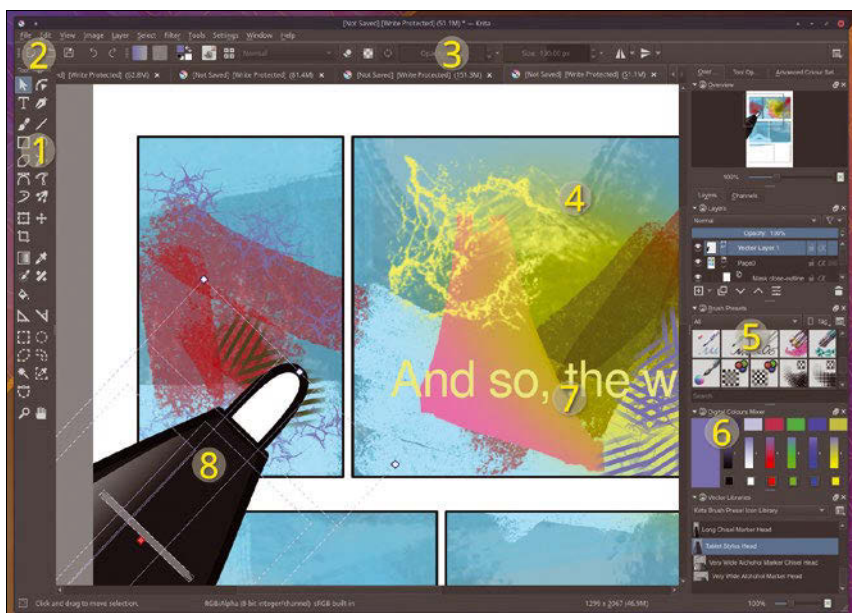
1. New vector tools: Drawing vectors is now easier, thanks to UI feedback. **2.** SVG by default: SVG is now the default file format, and files are saved transparently in the background. **3.** New Python scripting: Script enhancements are everywhere, from brushes to the new plugin manager. **4.** Colorize mask: Autofill regions by drawing a few colored strokes and letting Krita do the rest. **5.** Brush editor: There's a new brush editor, larger brushes, and new brush organization. **6.** New palette docker: There's also a new KPL palette. **7.** Text tool: The text tool has also been overhauled. **8.** Vector libraries: Drag and drop vector clipart from a huge collection of images.

**Project Website**

https://krita.org

YouTube viewer
FreeTube 0.2.0

YouTube videos have become ubiquitous. Whether it's for surreptitiously linking to the latest meme, or even for video embedded within serious news stories, you can't escape Google's video behemoth. And increasingly, you can't escape the advertising and snooping that goes on while you watch either. Which is why it's surprising there aren't more standalone YouTube viewers. Your favorite ad-blocking plugin will usually do the job for your browser, but often the quickest way to view outside of a browser is by dragging and dropping into something like VLC, and that often breaks Google's caching and playback.

FreeTube is a standalone app that gets around many of VLC's compatibility issues by being built around Electron and Chromium, although you wouldn't know this by launching the application. You see a simple interface that allows you to quickly see what's popular, add your own subscriptions, or search in the time-honored YouTube way. Results appear as a list of thumbnails, and these can be selected and viewed just as they can be in your browser. It does all this thanks to the YouTube HTTP API (and hard-coded API keys), with results via the all-powerful *youtube-dl* script. However, there's no advertising, and the author promises much greater privacy. More importantly, playback is perfect and offers all the same features as watching in a browser. A sepa-
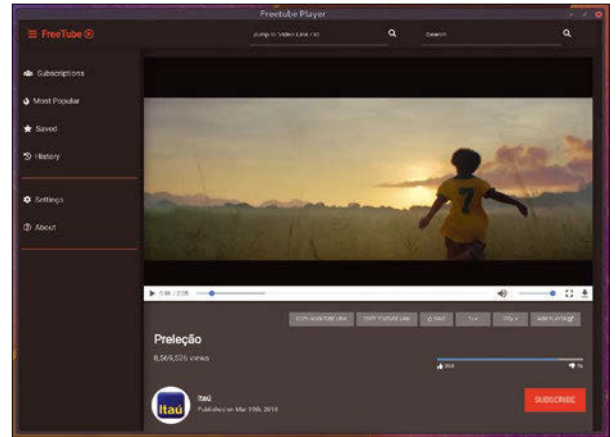
Unshackle your ASMR addiction from both the browser and those all-snooping Google cookies.

rate application like this makes a lot of sense, especially if you use a browser that isn't particularly compatible with YouTube (such as qutebrowser, below). This separation of tasks also has an advantage – web browsers have become a monopoly. With a separate app compatible with native shortcuts and subtitles, as well as an easy "save video" feature, there's almost no reason to browse to YouTube again.

Project Website
https://github.com/FreeTubeApp/FreeTube

Vim-binding browser
qutebrowser 1.2

With this release, qutebrowser has finally delivered on one of the original promises for the 1.0 milestone: the ability to create settings per-domain. This may seem like an esoteric feature, allowing you to change how qutebrowser behaves depending on which site it's browsing, but it's not when you start to think about how it could be used. You could change the keybindings for a specific site, for example, allowing you to reply to Reddit posts with your own specific command, or bind your favorite editor keys to an online editor. You could even use this option to disable JavaScript for specific sites and save that setting. What's even more powerful is that the commands used to configure a

domain take a pattern rather than a specific domain name, which means you can set general rules for a set of similar sites or domains, such as the whole of Facebook or Google. Scripts for the current host can also be toggled, and a simple press of the *S* key saves settings with subdomains included.

This is still an early implementation of what's hoped to be a more ambitious per-domain setting and plugin system, but it's still a powerful addition for this release. There's also more to this release than per-domain settings. You finally get Vim's visual/caret mode, for instance. You can use this to extend copy regions from the keyboard, such as when searching for a word. Now just press *v* and use the
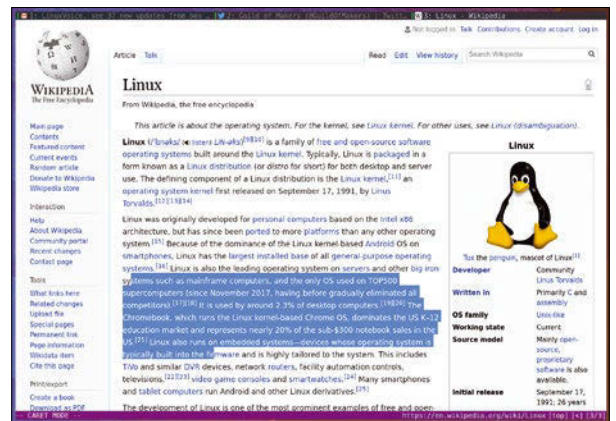
One of qutebrowser's best new features is its implementation of Vim's caret/visual mode.

cursor keys to extend the selection, potentially pressing *y* to yank that selection to the buffer. Another great addition is the special *qute://bindings* page, which simply shows all the keybindings configured for your browser, obviating the need to refer continually to the excellent qutebrowser cheat sheet many of us use.

**Project Website**
https://www.qutebrowser.org/

2FA generator
Authenticator

Many of us now use two-factor authentication (2FA) when connecting to vital online services such as Google Mail and GitHub. 2FA adds a second factor alongside your password, with the most common second factor being a time limited one-time pin generated by an app on your smartphone. Often this app is proprietary, such as Google's own Authenticator, but because old versions of the app were open source, there are also many "free" implementations that are capable of registering the same services and generating compatible codes. The best on Android, for example, is FreeOTP+ (the plus is important), because it allows you to back up your authentication credentials for each service, rather than having to re-en-

ter them every time you switch phones or devices.

But using a smartphone app isn't always ideal. This is where the simply named Authenticator helps. Authenticator is a Gtk3+ application that fits perfectly into a Gnome desktop and looks very similar to the glut of Android apps built after Google Authenticator. It's got both a light and dark theme and looks lovely, oriented in a portrait style much like the app. It's best feature is that when you press the *Add* button to add a credential, almost every service you can think of that supports 2FA is listed, rather than leaving you randomly entering encryption keys in the hope it works. With that done and with the necessary keys exchanged with the online service, you can then use Authen-
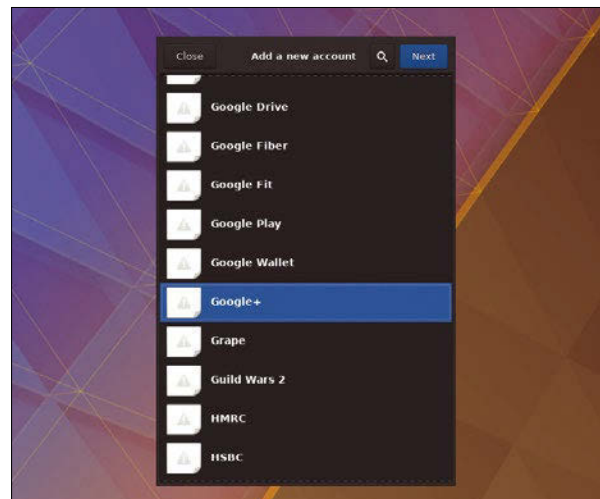
Authenticator can easily be installed using the `flatpak` command.

ticator just like the app – selecting the pre-configured service to get a time-limited code and entering this into your login prompt. Having this facility on the machine where you're likely to use the codes is very convenient. However, it does add a security risk, as anyone with access to your machine will be able to connect to your services (if they know the password), so be careful!

Project Website
https://github.com/bilelmoussaoui/Authenticator/

Firewall
OpenSnitch

Even though it's poorly implemented, Windows 10's dynamic firewall feature is theoretically a worthwhile addition. It notifies you when an application it doesn't know about, or an application without adequate permissions, attempts to make an outside network connection. You can then choose to enable it or block its attempts to communicate. Much like Android, this would be great if you could trust that nothing else was getting through without notifying you, but that sadly isn't the case. However, Mac OS users have a better option with a third-party tool called Little Snitch, which does basically the same thing, but because Little Snitch registers kernel extensions on a Unix-alike system, it seems a little more trustworthy.

What would be even better than all of these, of course, is a Linux application that tries to do the same thing – enter Open-Snitch. OpenSnitch is a firewall tool that notifies you when an unknown process attempts to make a connection and lets you decide whether to whitelist the app, block the app from making the connection, or deny or allow the connection for this time only. You might wonder why you'd need something like this when Linux network security is already very high, but it's because we often install and run so much that it can be difficult keeping abreast of what permissions are needed by everything. Open-Snitch does exactly this. It will automatically detect an outgoing connection regardless of the
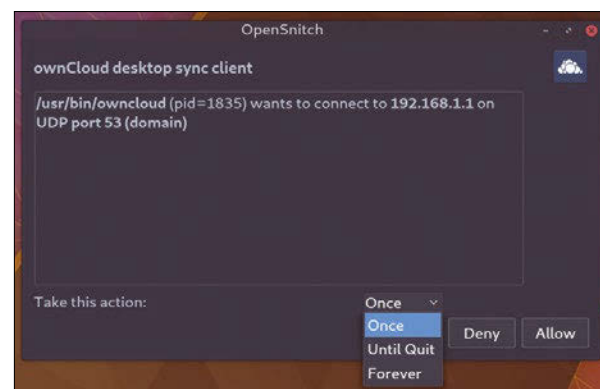
For the firewall functionality to work, OpenSnitch needs a kernel with iptables, NFQUEUE, and ftrace kernel support.

source, allowing you to create your firewall configuration as you need it. It's also a great way to see what even trusted applications are doing over the network. Of course, if you suffer some kind of vulnerability, the first thing it will try to do is talk to a server, which can be stopped with OpenSnitch before any communication takes place.

Project Website
https://github.com/evilsocket/opensnitch

IRC client
Irc it (ii)

The IRC protocol is supremely hackable and relatively easy to implement, mostly thanks to the protocol being basic and universal – it's almost as simple as sending character codes to a port and reading character codes from a port. It's perhaps for this reason that there are so many different IRC clients and implementations, from the all-powerful, like Irssi or WeeChat, to the experimental. Irc it (ii) definitely falls into the experimental category, because instead of presenting IRC as an interactive text parser, it turns IRC into a minimalist FIFO filesystem. This means that your IRC server connection appears as a directory tree, populated by server, channel, and nickname directories. At the end of every

path is an in file and an out file. The out file is where you'll find the messages for that channel, whereas text placed within the in file will be delivered to the channel. For example, typing

```
echo "/j #linuxvoice"> ⤶
    irc.freenide.net/in
```

will join the Linux Voice channel and create a new directory.

It's a unique take on IRC that could even have practical uses, especially if you use IRC for sharing files. But this filesystem layout may be equally useful for automation tasks or for processing input and output, such as through a natural language parser or speech synthesizer. The project site includes several example bots, as well as en-
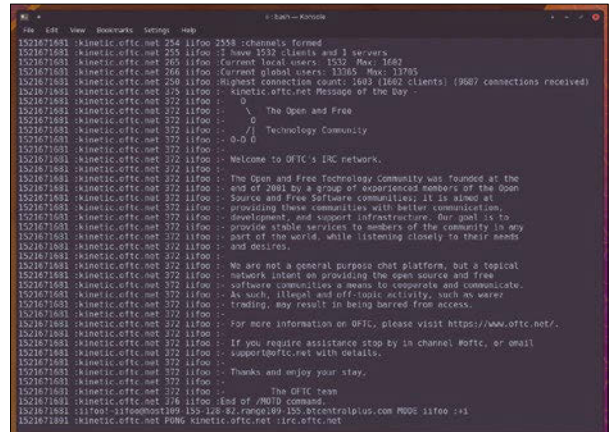
Turning IRC into a simple filesystem may initially sound like a gimmick, but it actually has many powerful and legitimate uses.

hancements that help you use the client from within Vim or as a PHP web front end. And while you're unlikely to find a pre-built package for your distribution, this project's tiny pocket of 500 lines of C code will build in less than a second and makes for a great template if you want to write you own IRC client.

Project Website
https://tools.suckless.org/ii/

Command-line assistant
Yoda

One of the first programs I wrote on a home computer was a personal assistant that held secret information about my family and friends and grew into a diary and calendar application, at least until the 27KB of RAM was used up (which doesn't take long when you're programming in BASIC). Yoda, a little command-line tool, reminds me of that old project, as it very much feels like a combination of tools that the developer has sewn together into a handy command-line utility. If you type `yoda hackernews`, for example, you can browse through the top news stories from that website directly from the terminal, which is useful because it obviates the temptation to dive into the comments for every story. Yoda is a useful

hotchpotch of ideas like these that also makes a great example Python project if you were looking to start your first programming project.

One of Yoda's neatest functions is a chatbot with which you can communicate, with natural language questions and answers parsed through an online API. A more useful feature is a speedtest that will quickly give you upload, download, and latency for your network connection. There's a URL shortener, too, which uses Google's shortening service. More substantially, there's a diary that includes the ability to add notes, create tasks, and jot down ideas. Additionally, there's a rather creepy mode where you can make notes about your "loved ones," such as their birth-
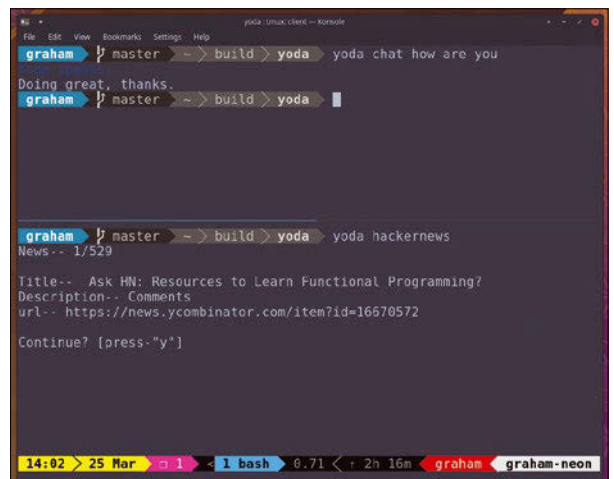
Ask your questions on the command line and get mystical answers, such as what the top story might be on Hacker News.

day and the things they like. Finally, there's even a money tracking mode for entering your expenses, plus flashcards for remembering things. It's obviously a disparate collection of tools, but there are some useful features, and it's a great first project to study or expand upon.

Project Website
https://github.com/yoda-pa/yoda

Web browser

Falkon 3.0

I f we're going to take back the web from the companies that are now defining what it has become, we need to own the browser. Fortunately, and quite unlike 15 years ago, all the major browsers have a major open source component, which means they can be ratified, forked, and expanded if necessary. And that's a good thing to do, because competition can help even the largest megacorp do the right thing. The recent major enhancements to Firefox are a good example, as Firefox is now defining its own terms for the web it wants to see – and that's something we can all support. But Firefox isn't the only open source web browser. We keep covering qutebrowser, for instance, because of its unique functionality, and Falkon is another worthy candidate. Like Phoenix that became Firefox from Netscape, Falkon is a new browser with old roots. It was formally QupZilla, a highly-regarded Qt-based web browser that was popular with KDE users (and us, as we looked at QupZilla in previous issues).

QupZilla became an official KDE project with the release of 2.2 in late 2017, and this positioning left the project ripe for a rebranding as it become an integral part of the KDE desktop. That rebranding is Falkon, and this is the first release, hence the 3.0 release tag (coming after QupZilla 2.2). If you're a KDE user, the first thing you'll appreciate is that Falkon completely respects your theme settings. This is the main problem with browsers like Chromium and Firefox, because even when there are matching themes for your KDE look, such as Arc Dark, the simple differences that come about from using different toolkits mean that it's almost impossible to make the browsers look completely integrated with your desktop. With Chromium, there's the added complexity of it using its own window decoration, as well as all the non-conforming settings pages and tab manipulation. But Falkon also has the option of looking like Chrome, and this may be especially useful if you're using a different desktop and would like some of Falkon's other features.

As you might anticipate, with Falkon now being a fully fledged KDE application, its main feature is that there are so many things
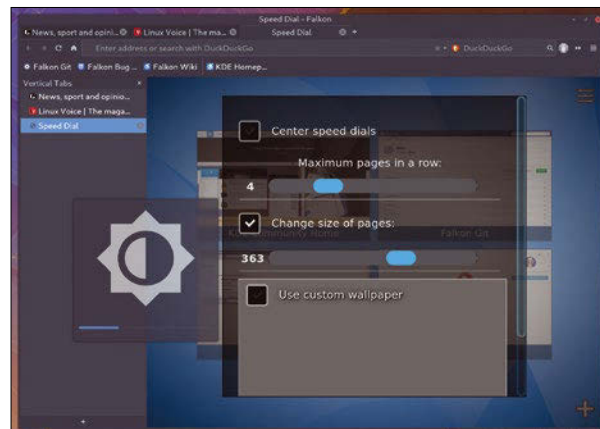
If you already use QupZilla, all your settings can be seamlessly migrated to Falkon by simply renaming the configuration directory to `falkon`.

you can change about the application. It installs with DuckDuckGo as the default search engine, and Adblock is installed and enabled when you dive into the configuration menus. Different sets of configuration options can be saved as a profile, and there are dozens of separate options for changing how both the tab bar and the URL bar work, as well as how downloads are handled. There's a decent integrated password manager that can be encrypted, although KWallet integration is a worthy extension, and the application supports native Linux and KDE notifications. "Extensions" allow you to augment this functionality with new plugins, and this release includes a new "vertical tabs" option that lists tabs down the left side of the application rather than horizontally across the top (although you can have both if you don't remove the horizontal bar). Thanks to the QtWebEngine, there's excellent web compatibility, and the desktop integration helps KDE users in particular if they're after a completely homogeneous desktop experience. It's certainly a lot better than using Konqueror for everything in the olden days, although I miss having Reddit open in the file manager while yet another pane is used to host the terminal, but maybe that will come.
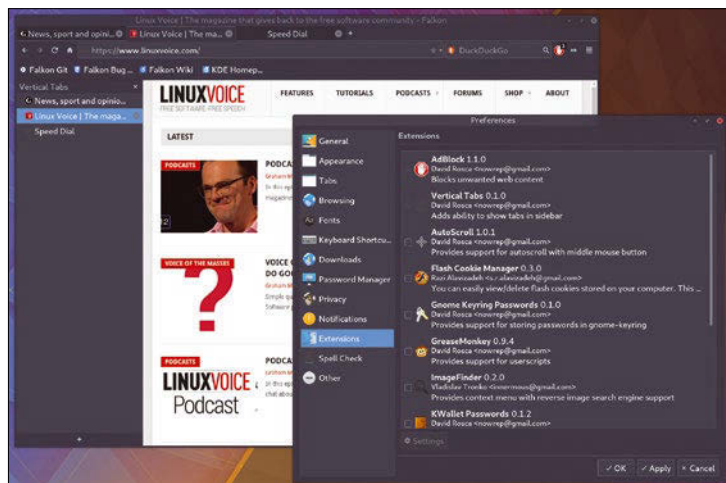
Not only is Falkon an excellent KDE-centric web browser, it's also one of the best Amiga flight sim games.

Project Website
https://github.com/KDE/falkon

City simulator
Citybound

It's understandable that so many different video game genres originated in the 1980s. This was the first time that computing had become affordable for millions of people and the first time that developers could make a living writing games. This blossoming of the games industry and the subsequent competition between programmers, designers, and publishers is what ultimately fueled the innovation. There was perhaps no better example of this than SimCity, a game that endeavored to simulate nothing less than the environment in which many millions of (US-based) players lived. But SimCity had its limitations. It wasn't really an emulation of the working environment of the cities you built, but rather a series of cleverly hidden factors and algorithms tailored to make the growth, pollution, and prosperity of your creations feel alive. To be fair, the hardware's processing capabilities at the time wasn't capable of much more, but that didn't stop Sid Meier when he augmented much of what SimCity hinted at with his amazing Civilization. And it's this quest to innovate the same genre that seems to be driving Anselm Eickhoff and his crazily ambitious Citybound. Citybound is aiming for nothing less than the tracking of every virtual inhabitant: how they prosper, grow, build families, and work. Your choices, as the city builder, affect their development and subsequently the evolution of your city. The game is
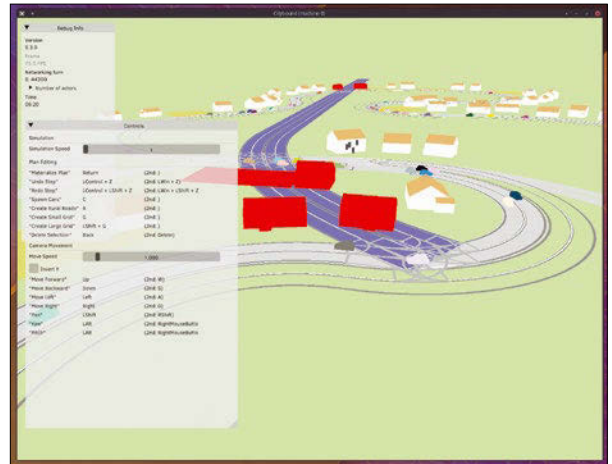
Citybound has been in development since 2014. While still only a prototype, it has a huge community of supporters and backers.

still very much a prototype, but as you drag the mouse around to create the environment, it becomes uncannily populated and suburbanized, and there are already many game elements with which to play. And if you want to help the project grow as much as your virtual city, you can support the developer on Patreon and join the already immensely popular community.

Project Website
http://cityboundsim.com/

Doom engine
GZDoom

Another title that's enjoyed remarkable longevity is Doom. Doom is about as far removed from SimCity (see above) as you could imagine (unless you include the giant lizard attack as one of SimCity 5's disaster scenarios), but it created a genre in much the same way. The first-person shooter (FPS) genre that Doom created is perhaps the most popular of all, with the latest release featuring hundreds of people on the same map at the same time, and graphics and physics that are sometimes indiscernible from reality. Yet the immediacy and relative simplicity of the original has kept it popular – popular enough that there are several open source implementations of its famous game engine, which itself was generously released for free in 1997. GZDoom is one of these, and it's actually a port of an older engine, now discontinued, called ZDoom.

The main advantages any of these recent engines will have are that they'll run on modern hardware and often use as much graphics acceleration as you can throw at them. GZDoom includes advanced rendering for both OpenGL and software-only platforms, which means you get advanced features like colored lighting, 3D floors, portals that connect separate areas of a map, and, of course, higher resolutions for both the display and the textures used within the maps. To play the original Doom with any new engine, you will need the

Likely using fewer polygons in the whole game than a single gun model in a modern FPS, Doom is still immensely satisfying.

original WAD files. As these have never been released under an open source license, you will need to extract them from your own personal copy of Doom. However, WAD files have been created and released under an open source license, with one of the best being "Freedom," a single-player game split into two parts and consisting of more than 60 different levels!

Project Website
https://zdoom.org/

Like Qlockwork – Use QML to build smart graphical applications

QML makes writing desktop applications a breeze, and you can later compile them into standalone programs that work more or less anywhere.

BY PAUL BROWN

In last month's issue, I looked at how to use a phone's GPS to live-stream a geographic location to a computer that didn't have a GPS [1]. Then, I took the whole thing on the move with an interactive map you could put together with QML [2], Qt's declarative scripting language.

In this article, I'll look at QML in more depth and build a graphical sample application from scratch. To get started, you can install the bits and pieces you need in Debian/Ubuntu-like distros with:

```
sudo apt install qt5-default ⏎
    qtdeclarative5-qtquick2-plugin
```

The *qt5-default* package pulls in all the Qt libraries you need, and *qtdeclarative5-qtquick2-plugin* brings in the tools you need to write QML applications.

In most distros, installing *qtdeclarative5-qtquick2-plugin* will also install QML's multimedia package, but if it doesn't, look for something like *qml-module-qtmultimedia* in your package manager and install that, too. QML's multimedia module contains all the necessary pieces to interface with your webcam and play video in a window.

Webcam

For this tutorial, you'll be making a webcam application with some functionality. Your app will end up being a bit like Cheese [3] or Kamoso [4], but with fewer bells and whistles. (Kamoso, incidentally, *is* built using Qt.)

The first thing you'll want to do is bring the webcam stream into a window. Listing 1 includes the modules you need for your application to work (lines 1-3). You will always need `QtQuick`, as well as `QtQuick.Window` if you are going to create graphical applications. The former contains all the basic code for QtQuick applications, and the latter has the libraries for creating and managing windows. The `QtMultimedia` module, on the other hand, contains a more specialized library of objects that manage video, audio, and, yes, the webcam on your computer.

Next, you open a `Window` (line 5). This construct is a QML *type* [5] that does what it says: It creates

a window. Everything that affects that window goes between its curly brackets (lines 5-17), including the name you want to appear in the window's title bar (line 6) and whether it is visible or not (line 7). Both `title` and `visible` are default `Window` attributes, or *properties* in QML parlance. Many other properties pertain to the window's size, position, opacity, background color, and so on. You can add your own properties to a QML type, too, as you will see later.

Line 9 declares a `Camera` type [6], and line 10 gives it a name. The `Camera` type construct comes with the `QtMultimedia` module. Unless you tell it otherwise (using a `Camera` property called `deviceId`), `Camera` uses your default webcam.

The `VideoOutput` type (line 13) also comes with the `QtMultimedia` QML module [7]. Its job is to show a video stream. The `VideoOutput` attribute `source` tells it where to get the stream. In this case, it is getting it from the `Camera` object (line 14). The source could also come from a video file.

The `anchors` property, which is common to most graphical elements in QML, makes sure the video feed fits inside the window. Anchors [8] are one of

Listing 1: WebCam v1

```
01 import QtQuick 2.9
02 import QtQuick.Window 2.2
03 import QtMultimedia 5.8
04
05 Window {
06  title: "WebCam"
07  visible: true
08
09  Camera {
10    id: camera
11  }
12
13  VideoOutput {
14    source: camera
15    anchors.fill: parent
16    }
17 }
```

the ways to position graphical objects. They allow you to place one object relative to another and set their size. In this case, you want the video "screen" to adapt (`fill` from side to side and/or top to bottom) to the size of the parent object (i.e., the object containing the video, or the `Window`). Because the `VideoOutput` object is anchored, if you resize the window, the video will resize with it.

Save the code as `webcam.qml` and then run it with the command:

```
qmlscene webcam.qml
```

The video feed does not actually fill the window (Figure 1) because the default `fillMode` is set to `"PreserveAspectFit"`, which scales the feed to fit the container while preserving its proportions. However, if you were to add the line

```
fillMode: "Stretch"
```

to the `VideoOutput` section, the video would distort to fit the window, and resizing the window will distort the feed further.

A third option for the `fillMode` property is `"PreserveAspectCrop"`, which scales uniformly by preserving the aspect ratio but cropping off the edges.

Impressively, only 17 lines of clear code gets you a window with a video stream from your webcam, which speaks volumes on how beginner-friendly QML is.

However, the window is tiny, and the app isn't very usable – you don't even have a button to close it. You can solve both problems by applying some layout and adding another element: a button.
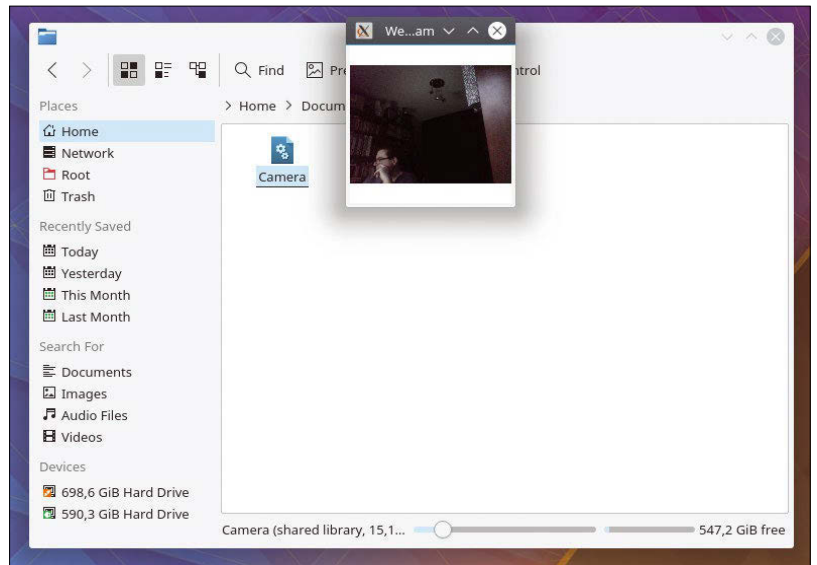
Figure 1: Your first (very basic) webcam application.

Multiple Elements

The QML type used to create a button is called, perhaps unsurprisingly, `Button`. At its most basic, it looks like the `Close` button built in Listing 2 (lines 33-35).

You will have noticed some other changes, of course: Because you have more than one graphical element in your window, you need to apply some layout elements to make sure everything is in its correct place; for example, so the button isn't placed on top of the video.

When you use `ColumnLayout` (lines 19-36), elements are stacked one on top of the other, as opposed to the `Row` or `RowLayout` types, which place objects next to one another. In this case, you place the `VideoOutput` element at the top (lines 26-30) and then a simple `Button` element below that (lines 33-35) with the `"Close"` label (line 34).

Listing 2: WebCam v2

```
01 import QtQuick 2.9
02 import QtQuick.Window 2.2
03 import QtMultimedia 5.8
04
05 import QtQuick.Controls 1.4
06 import QtQuick.Layouts 1.3
07
08 ApplicationWindow {
09   title: "WebCam"
10   visible: true
11
12   property int vpwidth: 640
13   property int margin: 10
14
15   Camera {
16     id: camera
17   }
18
19   ColumnLayout {
20     anchors.fill: parent
21
22     Item{
23       implicitWidth: vpwidth + margin * 2
24       implicitHeight: baseLayout.sourceRect.height * (vpwidth
        / baseLayout.sourceRect.width) + margin * 2
25
26       VideoOutput {
27         id: baseLayout
28         anchors.fill: parent
29         source: camera
30       }
31     }
32
33     Button {
34       text: "Close"
35     }
36   }
37 }
```

To get everything to work, you need to make a few extra tweaks, including the addition of two more modules: `QtQuick.Controls` (line 5) contains buttons, text boxes, radio buttons, and other elements, and `QtQuick.Layouts` (line 6) provides the layout tools used above.

Also notice that the containing graphical element has changed from `Window` to `ApplicationWindow` (line 8). The `ApplicationWindow` type provides more features than the plain old `Window` type. For example, you can easily implement a menubar and a status bar. More importantly for this project, though, it lets you adjust the window size better to the content.

Talking of size, you are going to set the size of your video screen this time around, rather than let the size of the window affect the content. To do this, you create two new properties: `vpwidth` (i.e., the video panel width; line 12) and `margin` (line 13), which will be a border around the video panel.

As for the height of the screen, you don't initially know the proportions of the video to be shown, so you can't hard-code it in. However, you can calculate it, because `VideoOutput` has a property called `sourceRect` that contains the dimensions of the video feed. If you establish the width of the feed as 640 pixels, as shown on line 12, you can use that to calculate the height of the container, as shown on line 24.

The container, by the way is an `Item` type. A QML `Item` is like an invisible frame in which you can enclose other graphical elements. In this case, it comes in handy because the `VideoOutput` type does not have the `implicitWidth` and `implicitHeight` properties needed to resize the panel – but `Item` does, which means you can set the size of the "screen" using the `Item` container (lines 23 and 24) and then use `anchors.fill: parent` to resize the video within the `Item` (line 28).

To summarize the changes:
- Lines 5 and 6 add two new modules: `QtQuick.Controls` for the button and `QtQuick.Layouts`

for the tools to organize elements within the window.
- Line 8 defines `ApplicationWindow` as your main container because it resizes more gracefully to the content.
- Lines 12 and 13 add two new properties: `vpwidth`, which holds the width you want for your screen, and `margin`, which holds the thickness of the border around the screen.
- Line 19 creates a `columnLayout`, which you expand to fill the whole window. This layout will hold the screen and the button, one above the other.
- Line 22 opens an `Item` container, which is an invisible frame to hold the video screen.
- Lines 23 and 24 set the size of the `Item` container.
- Line 27 names the `VideoOutput`, so you can refer to it (and get its size) elsewhere in the code.
- Lines 33-35 implement a button.

When you run the script, you'll see a much more reasonably sized application with a *Close* button (Figure 2), which doesn't do anything yet.

Adding Action

Up to now, you have added all the gears and cogs of the mechanism, but you haven't actually added the thing that makes everything move and work together. QML uses JavaScript to put a project's wheels into motion.

Apart from being declarative, QML is event driven, which means that the JavaScript executes when something happens (e.g., a button is clicked). In fact, that is what you are going to do here: Wait for the `Button` event `onClicked` [9] and then call the built-in routine `Qt.quit()` [10], which closes the current application.

The code could be:

```
onClicked: Qt.quit()
```

or:

```
onClicked: {Qt.quit()}
```

If you want to show a message on the command line about the application closing, you could do as in Listing 3. The takeaway is that `onClicked` is the declarative part of your code, and the bit after the colon is the imperative JavaScript.

As for where to place the code, you also have several options. You might be tempted to bang the code right in the `Button` section and that, as shown in Listing 3, would work fine.

However, for the sake of tidiness, modularity, and keeping the declarative part of your code separate from the imperative part, you might want to use the QML `Connections` type [11] (Listing 4).

With `Connections` (lines 47-50), you can give your active component (in this case, your button) an `id` (line 40) and then refer to it using the `Connections`

Figure 2: A proper webcam application with a Close button.
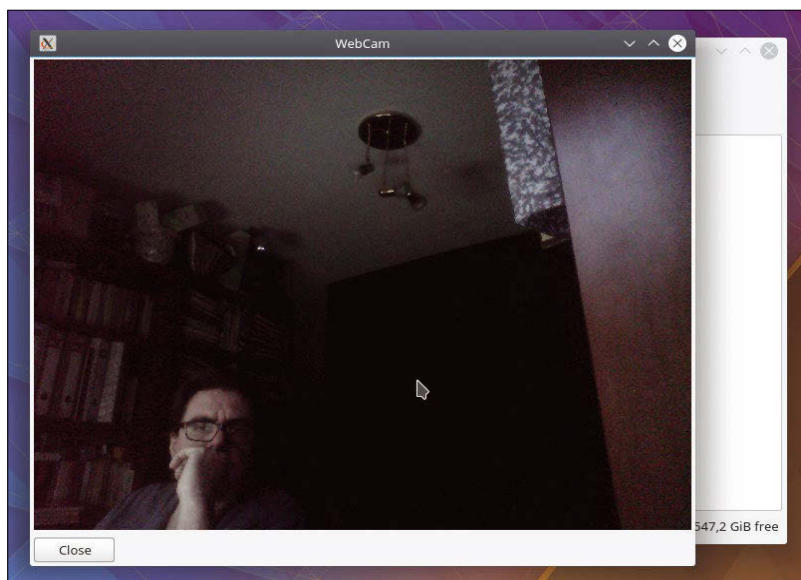
property `target` (line 52). The QML parser now knows to which component an event and its associated JavaScript refers.

You will notice another button in Listing 4: "Snap" (lines 34-37; Figure 3). When you click on *Snap*, it takes a snapshot of whatever `VideoOutput` is showing. You can see the code associated with the `snap` button on lines 47-50. The `Camera` subtype `imageCapture` allows you to take snapshots with its `capture()` method, as you can see on line 49.

The images are saved into the default directory or your system. In the case of Linux, that is the `Pictures` folder in your home directory.

Another property, `videoRecorder` [12] lets you record a video from the webcam. You might want to experiment with that, too.

Compiling

Now that your app is complete, it would be cool if you didn't have to invoke it with `qmlscene` every time but, instead, run it just like any other app. To do that, you need to compile it to binary, which will not only make it a standalone app but will also make it faster and more compact.

Before you start, make sure you have your basic compilation tools installed:

Listing 3: Close Button

```
[...]
Button {
 text: "Close"
 onClicked: {
  console.log ("WebCam App is closing...");
  Qt.quit();
 }
}
[...]
```

```
sudo apt install gcc g++ make
```

Then, you will need some Qt-specific packages:

```
sudo apt install qt5-qmake qtdeclarative5-dev
```

With the `qmake` tool, you will create a makefile that contains the information the compilers need to generate the binaries. The *qtdeclarative5-dev* package contains the libraries the compilers need to link against to generate binary code from the QML scripts.

Apart from your source code `.qml` files, you'll need to create three other files.

Listing 4: WebCam v3

```
01 import QtQuick 2.9
02 import QtQuick.Window 2.2
03 import QtMultimedia 5.8
04
05 import QtQuick.Controls 1.4
06 import QtQuick.Layouts 1.3
07
08 ApplicationWindow {
09  title: "WebCam"
10  visible: true
11
12  property int vpwidth: 640
13  property int margin: 10
14
15  Camera {
16   id: camera
17  }
18
19  ColumnLayout {
20   anchors.fill: parent
21
22   Item{
23    implicitWidth: vpwidth + margin * 2
24    implicitHeight: baseLayout.sourceRect.height *
       (vpwidth / baseLayout.sourceRect.width) + margin * 2
25
26    VideoOutput {
27     id: baseLayout
28     anchors.fill: parent
29     source: camera
30    }
31   }
32
33   Row {
34    Button {
35     id: snap
36     text: "Snap"
37    }
38
39    Button {
40     id: close
41     text: "Close"
42    }
43   }
44
45  }
46
47  Connections {
48   target: snap
49   onClicked: camera.imageCapture.capture ()
50  }
51
52  Connections {
53   target: close
54   onClicked: Qt.quit()
55  }
56 }
```
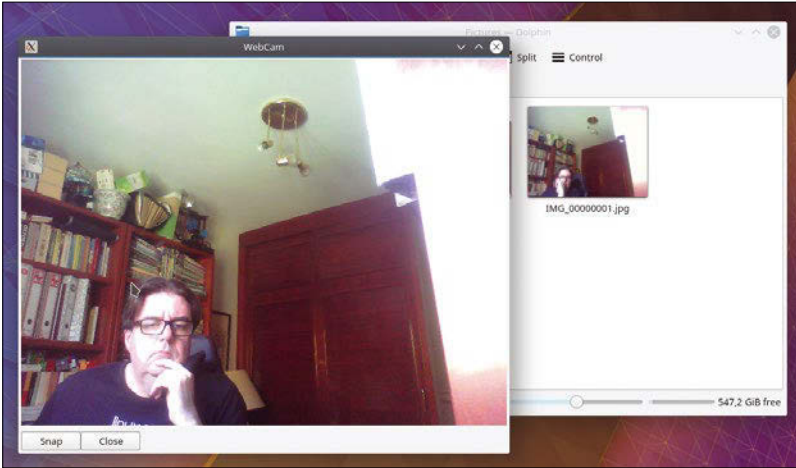
Figure 3: WebCam app with buttons that work, including the one that takes snapshots.

1. `qml.qrc` (Listing 5) contains the QML resources needed in the project (line 3). Because a QML application can contain various files (something I will cover in future installments) and span several folders, you state here what goes into the project and where it is located.
2. `webcam.cpp` (Listing 6) is the compiled C++ program. It is essentially a wrapper that loads several libraries and sets up an engine before calling in your `.qml` file (line 11).
3. `webcam.pro` (Listing 7) is the file `qmake` uses to generate the makefile. It adds the generic Qt tools `make` needs to be aware of (lines 1 and 2), as well as the files for your project (lines 3 and 4).

Once you have all your files, call `qmake` from the command line from within the directory containing

Listing 5: qml.qrc

```
01 <RCC>
02     <qresource prefix="/">
03         <file>webcam.qml</file>
04     </qresource>
05 </RCC>
```

Listing 6: webcam.cpp

```
01 #include <QGuiApplication>
02 #include <QQmlApplicationEngine>
03
04 int main(int argc, char *argv[])
05 {
06     QCoreApplication::setAttribute(Qt::AA_EnableHighDpiScaling);
07
08     QGuiApplication app(argc, argv);
09
10     QQmlApplicationEngine engine;
11     engine.load(QUrl(QStringLiteral("qrc:/webcam.qml")));
12     if (engine.rootObjects().isEmpty())
13         return -1;
14
15     return app.exec();
16 }
```

your files to generate a makefile. Next, run `make` to compile.

At the end of the process, you will see an executable file with the same name as your `.pro` file (in this case, `webcam`). That is your app, and you can run it from the command line or, in most graphical environments, by double-clicking it in your file manager.

Conclusion

Writing applications in QML is easy and, dare I say it, fun! Plus, although KDE's Plasma desktop is based on Qt, your apps will play really well with everything: Gnome, XFCE, Windows, Mac OS, and even Android. That means that the applications you write in QML will work on most desktops and, small screens allowing, on your smartphone.

In future installments, I will talk about developing for other platforms. Meanwhile, enjoy coding! ∎∎∎

Listing 7: webcam.pro

```
QT += quick
CONFIG += c++11
SOURCES += webcam.cpp
RESOURCES += qml.qrc
```

Info

[1] "Tutorials – gpsd" by Paul Brown, *Linux Pro Magazine*, issue 210, May 2018, pg. 90, *http://www.linuxpromagazine.com/Issues/2018/210/Tutorials-gpsd*

[2] Interactive map: *https://youtu.be/jcHOY07Rvpk*

[3] Cheese: *https://wiki.gnome.org/Apps/Cheese*

[4] Kamoso: *https://userbase.kde.org/Kamoso*

[5] QML Window type: *http://doc.qt.io/qt-5/qml-qtquick-window-window.html*

[6] QML Camera type: *http://doc.qt.io/qt-5/qml-qtmultimedia-camera.html*

[7] QML VideoOutput type: *http://doc.qt.io/qt-5/qml-qtmultimedia-videooutput.html*

[8] Positioning QML objects with anchors: *http://doc.qt.io/qt-5/qtquick-positioning-anchors.html*

[9] QML Button type: *http://doc.qt.io/qt-5/qml-qtquick-controls-button.html*

[10] Qt built-in quit() method: *http://doc.qt.io/qt-5/qml-qtqml-qt.html#quit-method*

[11] QML Connections type: *http://doc.qt.io/qt-5/qml-qtqml-connections.html*

[12] Camera videoRecorder property: *http://doc.qt.io/qt-5/qml-qtmultimedia-camerarecorder.html*
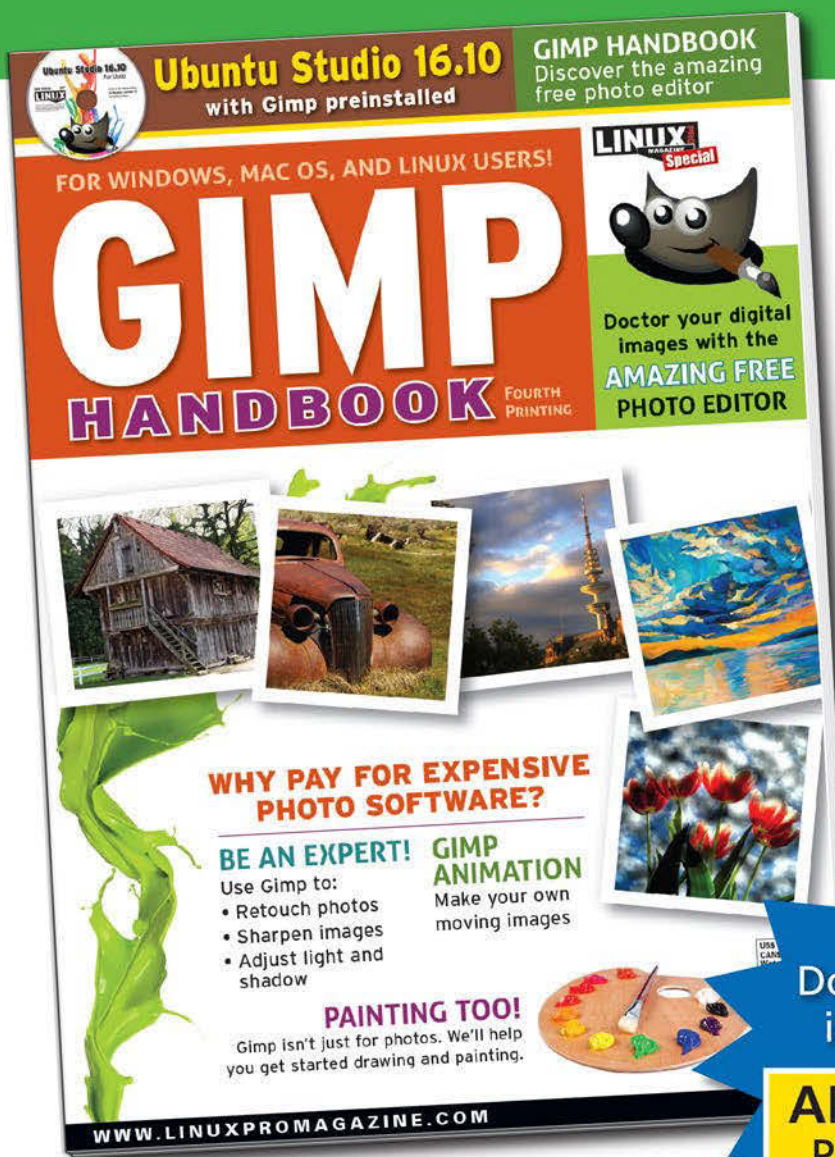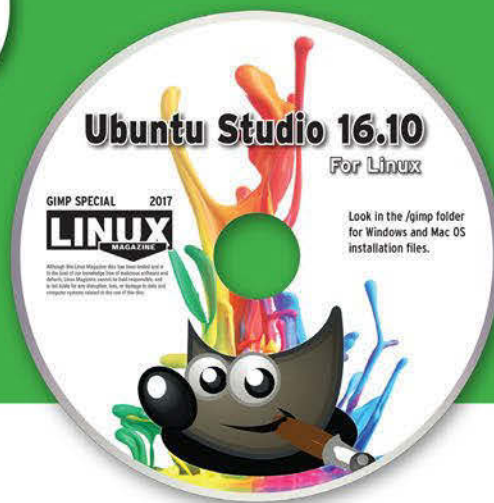
FEATURED EVENTS

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*.

Date: May 25–27, 2018

Location: Prague, Czech Republic

Website: *https://events.opensuse.org/ conference/oSC18*

The annual openSUSE community event that brings people from around the world together to meet and collaborate. The organized talks, workshops, and BoF sessions provide a framework around more casual meetups and hack sessions.

Date: June 8–9, 2018

Location: Austin, Texas

Website: *https://2018.texaslinuxfest.org/*

Texas Linux Fest is a community-run conference for Linux and open source software users and enthusiasts. Texas Linux Fest is a weekend event geared toward individual users. Whether you use free software and Linux at home, in your place of business, in your school or non-profit, or you are simply curious, Texas Linux Fest offers something for you.

Date: June 24–28, 2018

Location: Frankfurt, Germany

Website: *https://www.isc-hpc.com/*

ISC High Performance is dedicated to tackling HPC technological development and its application in scientific fields, as well as its adoption in commercial environments. This year the program team has selected 13 topics, which will be addressed in a number of thought-provoking sessions.

Events

openSUSE Conference 2018	May 25–27, 2018	Prague, Czech Republic	https://events.opensuse.org/conference/oSC18
Texas Linux Fest	June 8–9, 2018	Austin, Texas	https://2018.texaslinuxfest.org/
CeBIT 2018	June 11–15, 2018	Hannover, Germany	http://www.cebit.de/en/
Open Source Data Center Conf.	June 12–13, 2018	Berlin, Germany	https://osdc.de/
ISC High Performance 2018	June 24–28, 2018	Frankfurt, Germany	http://www.isc-hpc.com/
DebConf 18	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
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/
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

Erik Bärwaldt	22, 26
Swapnil Bhartiya	8, 14, 66
Paul Brown	90
Zack Brown	11
Bruce Byfield	48, 62
Claudio Cambra	16
Joe Casad	3
Mark Crutch	69
Karsten Günther	78
Jon "maddog" Hall	71
Gordon Horsington	42
Charly Kühnast	52
Christoph Langner	54
Martin Loschwitz	74
Vincent Mealing	69
Pete Metcalfe	58
Graham Morrison	84
Ferdinand Thommes	36, 72
Mike Schilli	32

Issue 212 / July 2018

Smart Home

The IoT revolution is blurring the line between domicile and device. Next month we look at some smart home technologies for Linux.

Preview Newsletter

The Linux Magazine Preview is a monthly email newsletter that gives you a sneak peek at the next issue, including links to articles posted online.

Sign up at: *www.linux-magazine.com/newsletter*

Lead Image © pol1978, 123RF.com

OPEN SOURCE DATA CENTER CONFERENCE

JUNE 12 - 13, 2018 | BERLIN

The Future of OS Data Centers

Be part of a unique Open Source event!

REGISTER NOW!

OSDC.DE