

FREE DVD

KaOS  
64-Bit

**4** COLLABORATIVE OFFICE SUITES:  
Keep your team in sync

**LIBREBOOT** and the liberation of firmware

LINUX  
MAGAZINE



# LINUX

**PRO**

## MAGAZINE

MAY 2018

# FREE BOOT!

## Libreboot and the liberation of firmware



### Electron

Web programming meets the desktop

### KNIME

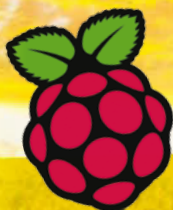
Powerful data analysis platform for non-programmers

### Ghost CMS

Is this free blogging tool better than WordPress?

### Bash Tricks

Manipulating binary data



### Raspberry Pi Slideshow

Beam out your message

# LINUXVOICE

- **NitroShare:** Easy cross-platform file exchange
- **Nootka:** Cool learning tool for teaching guitarists classical notation
- **Jasonette:** Create Android apps with minimal coding



### FOSS Picks

- MenuLibre
- Flameshot
- Chomper

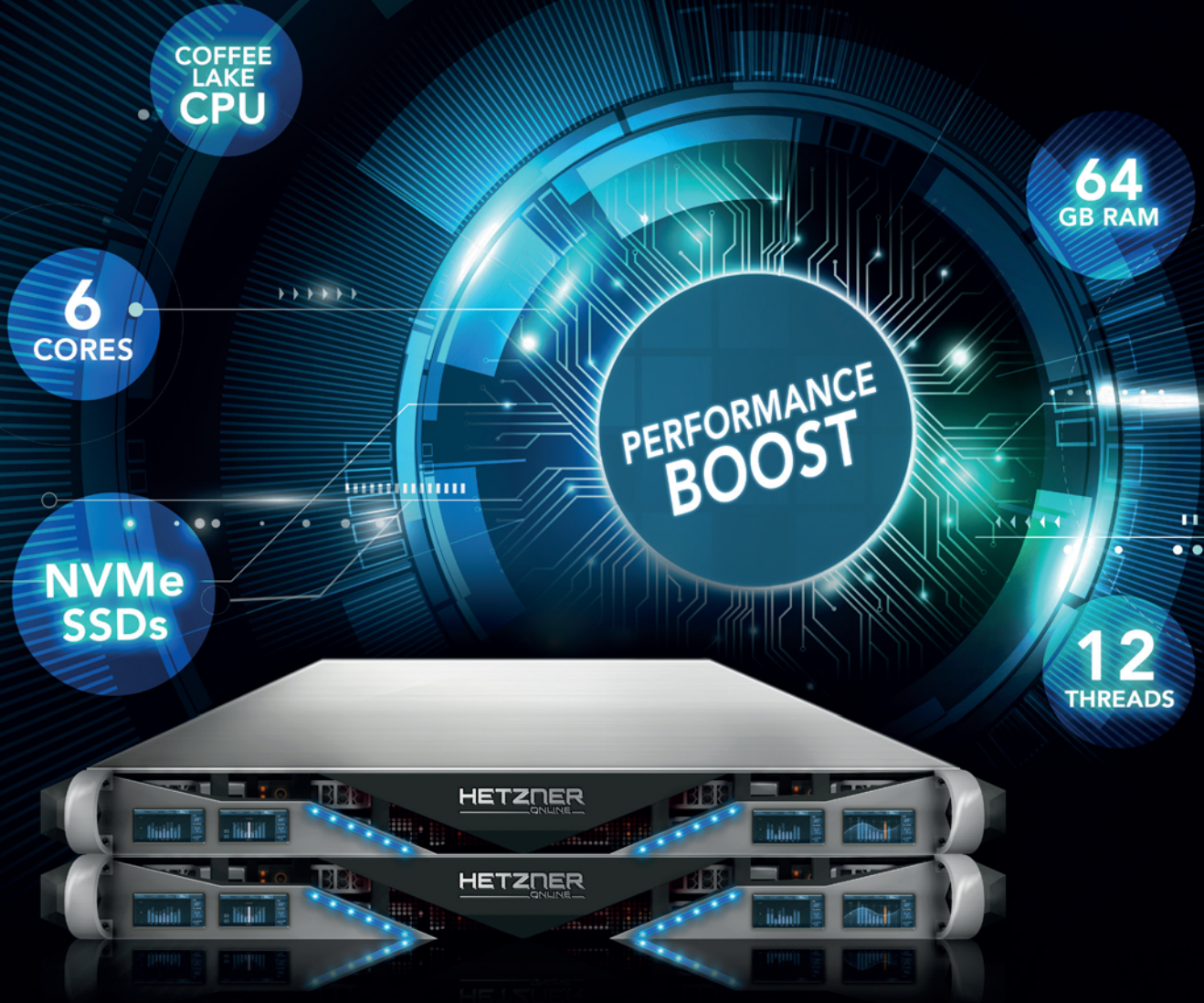
### Tutorial

Build your own GPS app

Issue 210  
May 2018  
US\$ 15.99  
CAN\$ 17.99



WWW.LINUXPROMAGAZINE.COM



### Dedicated Root Server EX61

Intel® Core™ i7-8700 Hexa-Core Coffee Lake  
incl. Hyper-Threading Technology  
64 GB DDR4 RAM  
2 x 6 TB SATA 6 Gb/s 7200 rpm HDD Class Enterprise  
100 GB Backup Space  
30 TB traffic inclusive\*  
No minimum contract  
Setup Fee \$147

monthly \$ **73**

### Dedicated Root Server EX61-NVMe

Intel® Core™ i7-8700 Hexa-Core Coffee Lake  
incl. Hyper-Threading Technology  
64 GB DDR4 RAM  
2 x 512 GB NVMe Gen3 x4  
100 GB Backup Space  
30 TB traffic inclusive\*  
No minimum contract  
Setup Fee \$147

monthly \$ **73**

[www.hetzner.com](http://www.hetzner.com)

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

All prices exclude VAT and are subject to the terms and conditions of Hetzner Online GmbH. Prices are subject to change. All rights reserved by the respective manufacturers.

# HATS OFF TO LARRY

Dear Reader,

We all loved it when Microsoft finally embraced Linux after a history of attacking it. Some of those outrageous sound bites like “Linux is a cancer...” got played back in triumphant retrospective as we all marked the passing of an era and remarked on the how far we’d come.

Should we be doing the same thing now with Oracle and the cloud? Oracle founder Larry Ellison made headlines back in 2008 with some very theatrical denunciations of the cloud industry. Ellison called the cloud “nonsense,” and once said of the cloud phenomenon, “Maybe I’m an idiot, but I have no idea what anyone is talking about. What is it? It’s complete gibberish. It’s insane. When is this idiocy going to stop?” [1]

Part of his argument was that much of what was getting called “cloud computing” was just a newfangled name for what used to be called *networking* or *web services* – a fair point at the time. But his vociferous refusal to acknowledge that anything was actually happening that resembled a new market or a paradigm shift seemed wildly controversial at a time when other companies were all rushing to reinvent and reorient. Many experts said that Ellison was just denouncing the cloud because his company was so far behind that it had no reasonable way to compete with cloud vendors other than to shame or frighten potential users away from using cloud services. (Actually, the same could be said about Microsoft’s attitude toward Linux.)

Fast forward to March of this year, and we discover that Oracle’s Q3 2018 report shows cloud revenues are up 32% to US\$1.6 billion. The press release includes a proud quote from Ellison, “The Oracle autonomous database is now fully available in the Oracle Cloud. And there are more autonomous cloud services to come. During this calendar year, we expect to deliver Autonomous Analytics, Autonomous Mobility, Autonomous Application Development, and Autonomous Integration services. Oracle’s new suite of Autonomous PaaS services delivers an unprecedented level of automation and cost savings to our customers.”

My they have come a long way! Or have they? At the time of Ellison’s anti-cloud tirade back in 2008, Oracle

was already working on its cloud portfolio, and they have labored systematically since then to transform their business around cloud services (just like everyone else is doing). If you were an observer from another planet, watching the behavior of Oracle the way Jane Goodall watched the behavior of wild chimpanzees, you would never have guessed that these people who were artfully executing this methodical entry into the enterprise cloud market worked for a company that regarded cloud computing as “complete gibberish.”

What is my point then? Oh, just a lament about this business I’m in, where you can interview these guys and they emphatically tell you one thing – with all the sincerity and bravado of a tenor in the opera – but all the while they are out there doing something different. Every tweet, every speech, every photo moment, is all about strategy, and *strategy* never stays long in the house of truth.

But I already knew that – and I guess you probably did to. Hats off to you, Larry. Sounds like things are on course, and your ship is sailing for another sunrise. So I guess maybe you *DID* understand the cloud after all.

*Joe*

Joe Casad,  
Editor in Chief



## Info

- [1] “Larry Ellison: Someone Explain to Me This Cloud Computing Thing”: <http://www.businessinsider.com/2008/9/larry-ellison-someone-explain-to-me-this-cloud-computing-thing-my-company-is-committing-to-orcl->

# LINUX MAGAZINE

## WHAT'S INSIDE

**This month** we tune into the Libreboot project – an effort to replace the proprietary firmware in laptop computers with an open source equivalent. Other highlights this month include:

- **Electron** – discover how to channel the cross-platform benefits of the web programming paradigm when writing desktop applications.
- **KNIME** – a graphical tool that lets you perform complex data analysis with custom workflows.

Also in these pages: check out MakerSpace for a tutorial on building a countdown timer with the Raspberry Pi, and jump to LinuxVoice to code your first Android app using Jasonette.

## SERVICE

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

## NEWS

### 08 News

- OpenStack Queens Released
- Kali Linux Comes to Windows
- Ubuntu to Start Collecting Data with Ubuntu 18.04
- CNCF Illuminates Serverless Vision
- Red Hat Decision Manager 7 Released
- Topological Superconductor Could Lead to Quantum Computing with Weird Fermions

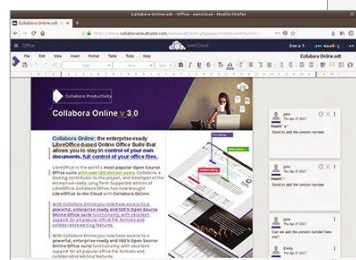
### 12 Kernel News

- Improving a Hashing Function
- Constant Values Adjustable at Boot Time
- Dealing with Intel Design Flaw

## REVIEWS

### 22 Collaborative Office Suites

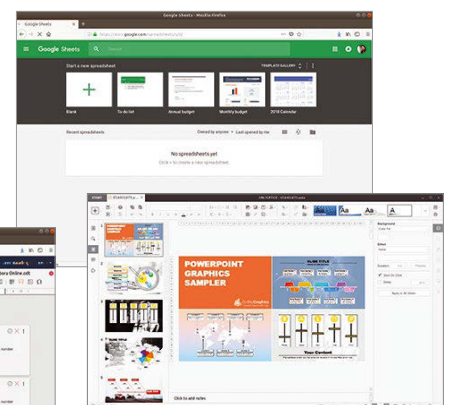
Several collaborative office suite providers are now competing for customers who are looking for the seamless exchange of office documents within an enterprise.



## COVER STORIES

### 16 Free Firmware with Libreboot

Libreboot is a project dedicated to making firmware free as in freedom.



**IN-DEPTH**

- 30 Electron**  
The Electron project brings the benefits of web programming to the realm of desktop applications.
- 34 Command Line – BorgBackup**  
In Star Trek, the Borg adds individuals to its collective consciousness, an apt metaphor for any backup application that stores individual files in an archive.
- 38 Programming Snapshot – Stock Quote API**  
Financial wizard Mike Schilli is annoyed that some web services have stopped serving up real-time stock market data. In order to keep an eye on his investment dollars, Mike taps into a little known interface for stock prices.
- 42 Ghost CMS**  
Blogging usually implies WordPress, but Ghost presents a serious challenge.



- 45 Charly's Column – sshuttle**  
When he doesn't want to deal with OpenVPN version conflicts or congestion control problems during TCP tunneling, Charly catches a ride on sshuttle.
- 46 KNIME**  
They say data is "the new oil," but all that data you collect is only valuable if it leads to new insights. An open source analysis tool called KNIME lets you analyze data through graphical workflows – without the need for programming or complex spreadsheet manipulation.
- 52 Binary Data in Bash**  
Bash is known for admin utilities and text manipulation tools, but the venerable command shell included with most Linux systems also has some powerful commands for manipulating binary data.

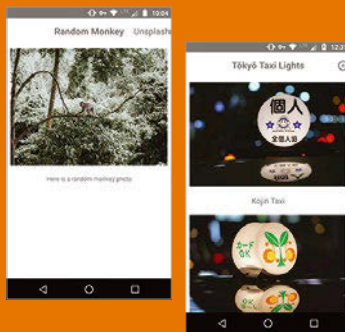
**MAKERSPACE**

- 56 Countdown Counter**  
Build a countdown counter with a Raspberry Pi and the MAX7221 display driver.
- 60 Open Hardware – Keyboardio**  
The Keyboardio Model 01 is a highly-customizable open hardware keyboard.
- 64 Raspberry Slideshow**  
Video sequences or slide shows promote sales. With a Raspberry Pi and the Raspberry Slideshow distribution, you can capture the attention you need.



**LINUXVOICE**

- 69 Welcome**  
Going cross-platform.
- 70 Jasonette**  
Build Android apps with a minimum of coding.
- 75 Doghouse – Subutai**  
Be a catalyst and help others break the bonds of proprietary software.
- 76 NitroShare**  
This handy app simplifies the exchange of files and directories between computers with Linux, Mac OS, Windows, and Android.
- 80 Nootka**  
With Nootka and your guitar, you can learn to recognize musical notes in classical notation.
- 84 FOSSPicks**  
Graham checks out VLC 3.0, MenuLibre 2.1.5, Texttop, Flameshot, Chomper, Godot 3.0, and much more!
- 90 Tutorial – GPSD**  
Even though your computer does not come with its own GPS, you can hack one onto it using a mobile phone – if you are ready for some fiddling.



# On the DVD

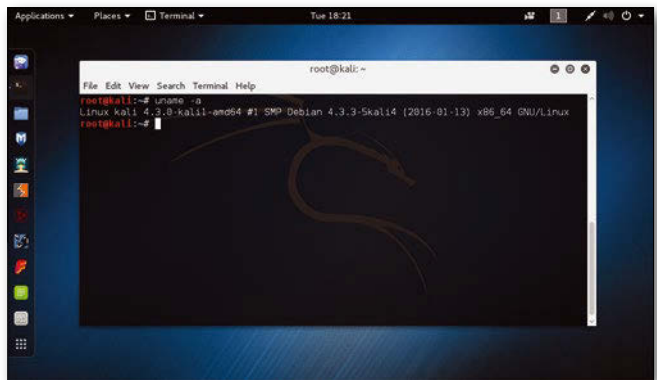


## Kali Linux

Kali puts the emphasis on penetration testing and security auditing. According to the developers, Kali comes with more than 300 security tools, including the Metasploit pen testing framework, as well as an assortment of utilities for port scanning, traffic analysis, password cracking, and more. The latest release comes with the Linux 4.14 kernel and includes increased memory limits and support for AMD secure memory encryption.

## KaOS

This lean Linux is a showcase for the KDE desktop and the Qt development framework. The KaOS developers' goal is to create the most usable, stable, and sensible KDE environment with a limited size to keep the emphasis on quality rather than quantity. Development efforts are geared toward packaging and perfecting the desktop environment, rather than adding new tools and applications.

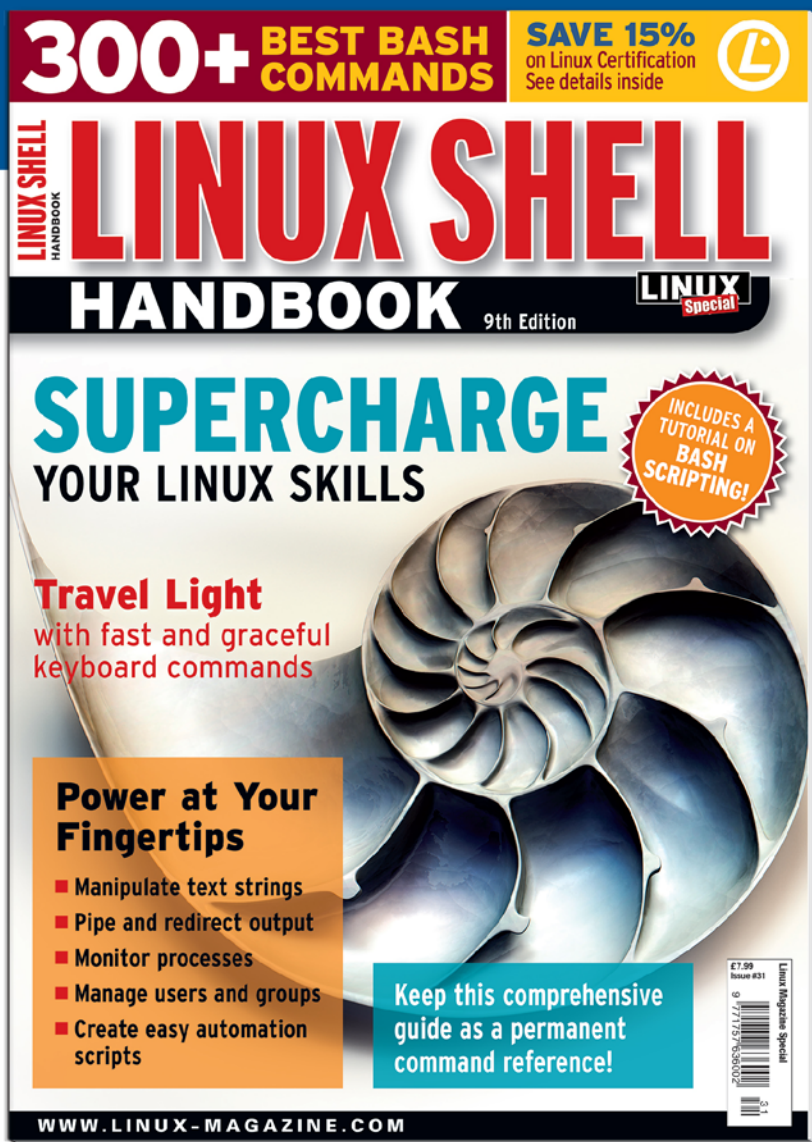


## Additional Resources

- [1] Kali Linux: <https://www.kali.org/>
- [2] Kali documentation: <https://www.kali.org/kali-linux-documentation/>
- [3] KaOS: <https://kaosx.us/>
- [4] KaOS documentation: <https://kaosx.us/docs/>

*Defective discs will be replaced. Please send an email to [subs@linux-magazine.com](mailto:subs@linux-magazine.com).*

# EXPERT TOUCH



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

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

You'll learn how to:

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

**9th Edition!**

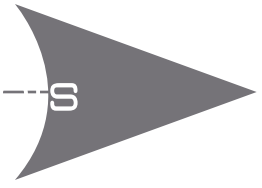
The best way to stay in touch with your system is through the fast, versatile, and powerful Bash shell. Keep this handy command reference close to your desk, and learn to work like the experts.

ORDER ONLINE:

shop.linuxnewmedia.com/specials

# NEWS

Updates on tech



## THIS MONTH'S NEWS

### 08 New OpenStack Cloud Features

- OpenStack Queens Released
- Kali Linux Comes to Windows

### 09 Linux Desktop Data Collection

- Ubuntu to Start Collecting Data with Ubuntu 18.04
- CNCF Illuminates Serverless Vision
- More Online

### 10 Automating Business Decisions

- Red Hat Decision Manager 7 Released
- Topological Superconductor Could Lead to Quantum Computing with Weird Fermions

## OpenStack Queens Released

The OpenStack Foundation has announced the 17th release of OpenStack, dubbed Queens. OpenStack now powers some of the biggest public clouds in the world. According to a 451 Research survey, 60 percent of enterprise workloads will run in the cloud. The survey also reports that enterprise adoption of OpenStack is expanding in parallel, with enterprises in nearly all verticals and regions now running mission-critical workloads on OpenStack software.

Queens is well equipped with many new features to handle such workloads. It comes with Cinder multi-attach that enables operators to attach the same Cinder volume to multiple virtual machines (VMs). If one node goes down, the other takes over and has access to the volume. According to the OpenStack Foundation, this was one of the most-requested features in cloud environments.

The Queens release also includes significant enhancements to support emerging use cases such as machine learning, network functions virtualization (NFV), container integration and edge computing.

"Our developer community keeps building new tools that are helpful to the infrastructure operator. Eight years in, features that enhance manageability, resiliency, scalability and user experience are still paramount and still have sizzle factor," said Jonathan Bryce, Executive Director of the OpenStack Foundation.



Image © satori, 123RF.com

## Kali Linux Comes to Windows

Kali Linux, a penetration testing distro that you may have seen in *Mr. Robot*, is now available in the Microsoft Store. The Kali Linux team has been working with the Microsoft Windows Subsystem for Linux (WSL) team to bring the distro to the platform that still dominates the PC landscape. By doing so, Kali has brought some of the best penetration testing tools to the biggest PC user base.

"This is especially exciting news for penetration testers and security professionals who have limited toolsets due to enterprise compliance standards," wrote Mati Aharoni, lead Kali developer, in a blog post.

At the end of last year, Microsoft took WSL out of beta, making it available for every Windows 10 user for the latest Fall Creators Update. Users can simply go to the Microsoft Store and install their preferred Linux distro just like any other app. Initially only Ubuntu, openSUSE Leap, and SUSE Linux Enterprise were available, but the WSL team is working with different distros to bring them to the store.

Officially only command-line Linux utilities are available through WSL, as the target audience is developers and sys admins who want to write or deploy applications for



Linux machines running on the cloud. That didn't stop the Linux community from running GUI applications through WSL. In fact, even the Kali Linux team managed to run a full blown XFCE desktop.

Running Linux distros under WSL has its own shortcomings. Your system is only as secure as Windows 10. However, the Kali Linux team sees some great possibilities too, "While running Kali on Windows has a few drawbacks to running it natively (such as the lack of raw socket support), it does bring in some very interesting possibilities, such as extending your security toolkit to include a whole bunch of command-line tools that are present in Kali."

Being able to run Linux inside of Windows 10 gives developers native access to tools from both worlds.

## Ubuntu to Start Collecting Some Data with Ubuntu 18.04

Canonical, the parent company of Ubuntu, is planning to collect diagnostic data from its desktop operating system. In a message posted to the Ubuntu Developer mailing list, Will Cooke, director of Ubuntu Desktop, explained the reason behind this move, "We want to be able to focus our engineering efforts on the things that matter most to our users, and in order to do that, we need to get some more data about what sort of setups our users have and which software they are running on it."

Ubuntu installer will have an opt-out checkbox for sending diagnostics information to help improve Ubuntu.

Canonical has chosen to keep this feature opt-out, instead of opt-in, meaning unless you uncheck the box, Canonical will collect diagnostic data. The Ubuntu privacy policy will be updated to reflect this change. In order to give users more control over the features, there will be an opt-out option in the Gnome System Settings.

What kind of data will Canonical be collecting? Nothing invasive. They want to know which flavor and version of Ubuntu you are running and whether you have network connectivity (one may wonder how will they get the data if there is no network connectivity?).

They will also collect data about the processor, GPU, screen resolution, memory, storage, and OEM manufacturer. Other data includes location (not IP address), installation duration, status of auto login, and disk layout.

Cooke said that all of this data will be made public. It could be a great way for Canonical to start collecting stats about the Linux desktop. There are no credible stats about who is using the platform. Canonical's move can be a step in that direction.



Image © Tatiana Venkova, 123RF.com

## CNCF Illuminates Serverless Vision

Serverless or Function as a Service is one of the hottest topics these days. But what is "serverless computing" and who is it for? Can it replace the existing models? These are some of the many questions the Cloud Native Computing Foundation (CNCF) is attempting to answer in a paper drafted by the CNCF Serverless Working Group.

"Serverless is a natural evolution of cloud-native computing. The CNCF is advancing serverless adoption through collaboration and community-driven initiatives that will enable interoperability," said Chris Aniszczuk, COO, CNCF.

According to the whitepaper, "Serverless computing refers to the concept of building and running applications that do not require server management. It describes a finer-grained deployment



Image © Galina Peshkova, 123RF.com

## MORE ONLINE

### Linux Magazine

[www.linux-magazine.com](http://www.linux-magazine.com)

### ADMIN HPC

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

#### What to Do with System Data: Think Like a Vegan • Jeff Layton

What do you do with all of the HPC data you harvested as a lumberjack? You think like a Vegan.

### ADMIN Online

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

#### Identity Management from the Cloud Martin Kuppinger

Offers for identity management as a service (IDaaS) are entering the market and promising simplicity. However, many lack functionality, adaptability, and in-depth integration with existing systems. We look at how IT managers should consider IDaaS in their strategy.

#### LibreOffice in the Workplace

Thomas Krumbein

How can LibreOffice be used in a corporate environment and what difficulties have to be considered? In particular, you cannot underestimate acceptance by your employees.

#### Regular expressions and metacharacters in PowerShell • Thomas Bär and Frank-Michael Schlede

Almost all programming and scripting languages allow the use of regular expressions, but many professionals still believe regex is a relic from ancient times. With specific examples, we show how useful and meaningful regex can be in PowerShell.

### ADMIN DevOps Focus

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

#### Automated Jenkins CI • Nathan Bliss

A Jenkins build pipeline frees developers from repetitive manual build tasks by facilitating automation and provides broad traceability with continuous logging and feedback.

model where applications, bundled as one or more functions, are uploaded to a platform and then executed, scaled, and billed in response to the exact demand needed at the moment.”

Being a new technology, there is a lot of work to be done for the healthy growth of serverless ecosystem. The CNCF has recognized its role in the space and is attempting to address those needs. The CNCF will start a drive to encourage more serverless technology vendors and open source developers to join the CNCF. It will also look at ways to foster an open ecosystem by establishing interoperable APIs, ensuring interoperable implementations with vendor commitments and open source tools.

CNCF is a Linux Foundation Collaborative project that was created to foster innovation in the cloud native space. Kubernetes was its anchor project.

You can read the whitepaper on GitHub.

## Red Hat Decision Manager 7 Released

Red Hat has announced the release of Red Hat Decision Manager 7, an open source product that makes it easier for businesses to automate business decisions.

If you want to book a flight, several variables might affect the cost. An airline or a travel agent platform needs rules that reflect those variables. Traditionally, those rules were created by business operators and implemented by software developers. Red

Hat Decision Manager 7 eases that process by automating it.

According to industry analyst firm IDC, non-traditional developers are expected to build 20 percent of business applications and 30 percent of new application features by 2021.



**redhat**

Business processes play an important role in helping organizations improve efficiency and reduce workloads, and at the heart of every business process are business rules.

Red Hat Decision Manager 7 is a low-code development tool that enables businesses to take a more active role in application development by creating an environment for collaboration between business and IT professionals. Red Hat Decision Manager 7 is built for both traditional and cloud-native applications.

## Topological Superconductor Could Lead to Quantum Computing with Weird Fermions

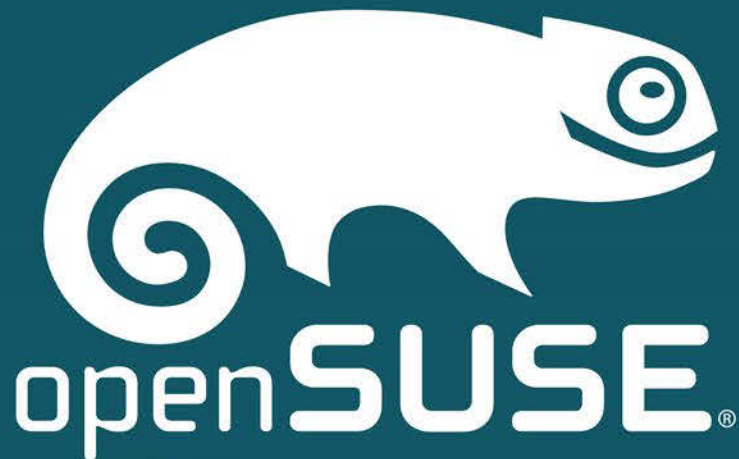
With their natural insensitivity to decoherence, Majorana particles, which are also called Majorana fermions, have long been considered a promising ingredient for quantum computing, but the difficulty in obtaining and containing the illusive particles has limited their usefulness in quantum research. A team at Chalmers University of Technology in Sweden has announced that they have developed a means for corralling Majorana particles using topological superconductivity.

According to the announcement at Physics.org, “Majorana fermions are highly original particles, quite unlike those that make up the materials around us. In highly simplified terms, they can be seen as half-electron. In a quantum computer, the idea is to encode information in a pair of Majorana fermions separated in the material, which should, in principle, make the calculations immune to decoherence.”

According to team leader and Chalmers professor Floriana Lombardi, “For practical reasons, the material is mainly of interest to those attempting to build a topological quantum computer. We want to explore the new physics hidden in topological superconductors—this is a new chapter in physics.”



Prague, Czech Republic



# Conference

Open Source Software

- ◆ Embedded Systems
- ◆ Cloud ◆ Containers ◆ More

May 25 - 27, 2018

[events.opensuse.org](https://events.opensuse.org)

# Zack's Kernel News



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

By Zack Brown

## Improving a Hashing Function

Amir Goldstein posted a small fix to a kernel function that would produce a unique hash value for an input string. He noticed that for 64-bit values, some bits were lost before calculating the hash – resulting in a slightly less good hash. For 32-bit values, he said, no bits were lost, and the code was left unchanged, but since the relevant `stringhash.h` file had no official maintainer, and Linus Torvalds had done some work on it in the past, Amir asked Linus what he thought of the fix and for advice on how to test it. As Amir put it, “I wouldn’t even know where to begin testing its affects, or how to prove if there really is a problem.”

Linus dug back into his memory, saying he believed the lost bits were intentional, though he acknowledged, “it’s a long time ago, and we’ve changed some of the hashing since.”

But Linus also said, “you’re wrong that it’s a no-op on 32-bit. It’s a very expensive and pointless multiplication there too, even if the *shift* ends up being a no-op. The name hashing is pretty performance-sensitive.”

Amir corrected Linus, saying that he hadn’t said the patch would result in a no-op (code that literally did nothing, not even use time); he’d said that the patch left the code for the 32-bit case unchanged, except for the bit shift Linus had mentioned.

Linus pointed out that while the shift might cost nothing, the `__hash_32()` used nearby was very expensive. So, he con-

### 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**.

cluded, “the patch as-is doesn’t seem to buy anything, and only adds cost.” He also remarked, “we did have numbers at one point. That’s what really matters: how good the actual hashing ends up being. So for me to take the patch, I would need to see that it actually improves the hash bucket spreading enough to be worth the cost.”

Amir replied that it was not his patch that added the `__hash_32()` call at all – it was already there.

To which Linus replied, “Oh, duh. My bad. It was indeed there already. Let me go back and look at the history of this thing.”

A few days later, he wrote back more fully:

*After having looked more at it, I take back all my complaints about the patch; you were right, and I was misreading things or just being stupid.*

*I also don’t worry too much about the possible performance impact of this on 64-bit, since most architectures that actually care about performance end up not using this very much (the `dcache` code is the most performance-critical, but the word-at-a-time case uses its own hashing anyway).*

*So this ends up being mostly used for filesystems that do their own degraded hashing (usually because they want a case-insensitive comparison function).*

*A `_tiny_` worry remains, in that not everybody uses `DCACHE_WORD_ACCESS`, and then this potentially makes things more expensive on 64-bit architectures with slow or lacking multipliers even for the normal case.*

*That said, realistically the only such architecture I can think of is PA-RISC. Nobody really cares about performance on that; it’s more of a “look ma, I’ve got warts^W an odd machine” platform.*

*So I think your patch is fine, and all my initial worries were just misplaced from not looking at this properly.*

*Sorry.*

Linus also remarked, “I *would* love to get some kind of estimate on how this changes the hash distribution. Do you have anything like that? The way this function is designed to be used, the upper bits should be used for the hash bucket information, so doing some bucket size distribution on (say) the upper ~12 bits or something with some real string input would be really good.”

Amir replied, “I’ve added bucket counters and tracing number of entries/number of non-empty buckets and largest bucket to dcache (patch attached). Increased `d_hash_shift` to take only 12 bits for hash table offset. The results vary a bit per run, but below are some outputs from sample runs of `find/` of 55K entries right after boot. The bottom line is that, at least w.r.t this simplistic bucket analysis and this specific workload, the patch does not seem to move the needle. The bucket distribution of before and after patch are similar and also similar to the word-at-a-time results.”

He included some numbers, and the thread ended.

## Constant Values Adjustable at Boot Time

Some parts of the kernel run so hot that even the overhead of assigning a value to a variable can result in a significant speed difference. Kirill A. Shutemov recently posted a patch to introduce “patchable constants.” These are constant values that can be set at boot time, either as part of the kernel configuration itself or by the user at the bootloader command line. Once set, they thereafter behave as true constant values in the running kernel, rather than variables.

As Kirill explained it, “Patchable constants implemented by replacing a constant with call to inline function that returns the constant value using inline assembler. In inline assembler, we also write down into separate section location of the instruction that loads the constant. This way we can find the location later and adjust the value.”

The reason Kirill wanted to implement this general feature was to solve the specific case of the `__PHYSICAL_MASK` variable. To do memory encryption, this had to be dynamically settable. But because memory encryption has a speed impact on absolutely everything, Kirill wanted to

find a way to shave as many Assembly instructions off the code as possible. Reimplementing the variable as a patchable constant solved the problem.

However, Kirill acknowledged that it came with an unacceptable cost. He said, “This conversion makes GCC generate worse code. Conversion `__PHYSICAL_MASK` to a patchable constant adds about 5k in `.text` on `defconfig` and makes it slightly slower at runtime (~0.2% on my box).”

So, the very effort to speed things up in one way, ended up slowing down the code in another. Kirill pointed out that he was submitting this patch, not to be included in the kernel, but in the hopes that someone might have a better idea for how to get the benefits of patchable constants, without this unwanted cost.

Linus Torvalds replied, saying:

*I actually wanted something very close to this, but I think your approach is much too simplistic. You force all constants into a register, which means that the resulting code is always going to be very far from non-optimal. You also force a big `movabsq` instruction, which really is huge, and almost never needed. Together with the “use a register,” it just makes for big code.*

He added, “What I wanted was something that can take things like a shift by a variable that is set once and turn it into a shift by a boot-time constant. Which means that you literally end up patching the 8-bit immediate in the shift instruction itself.”

He posted some assembly code, showing how several lines could logically be boiled down to just one, which he said would result in “much smaller code, and register `%rcx` isn’t used at all. And no `D$` miss on loading that constant (that is a constant depending on boot-time setup only).” He went on to say, “It’s rather more complex, but it actually gives a much bigger win. The code itself will be much better, and smaller. The *“infrastructure”* for the code gets pretty hairy, though.”

He pointed to a discussion from 18 months earlier, in which H. Peter Anvin had done some work on a similar idea. Peter had called it a “disgusting pseudo-self-modifying code idea,” and it hadn’t been completed because of all the extra complexity it introduced.

Peter replied, “I am currently working on it much more comprehensive set of patches for extremely this. I am already much further ahead and support most operations.” He added, “The patchset I have is about 85% complete. It mostly needs cleanup, testing, and breaking into reasonable chunks.” He also remarked, “The main reason I haven’t submitted it yet is that I got a bit overly ambitious and wanted to implement a whole bunch of more complex subcases, such as 64-bit shifts on a 32-bit kernel. The win in that case is actually quite huge, but it requires data-dependent code patching and not just immediate patching, which requires augmentation of the alternative framework.”

The discussion ended at that point.

Because open source is so decentralized, overlapping projects like this do sometimes happen, often resulting in a lot of wasted work or ongoing controversy over which approach is better. But in this case, given that Kirill had posted his patch specifically because he had not found a way to solve the underlying problems, it doesn’t look like there will be much controversy, at least over which project to accept into the kernel. There still may be some debate over the value of adding so much complex infrastructure to handle self-modifying constants. It may turn out to be so bug-prone and unmaintainable that it defeats the value of the speed increase.

## Dealing with Intel Design Flaw

Recently, a serious design flaw with Intel microprocessors has forced various operating systems, including Linux, to come up with some workarounds. Recently, KarimAllah Ahmed posted a patch to control Indirect Branch Speculation (IBS), essentially trying to manage the hardware deficiencies of Intel chips, to activate them when needed and safe, and to lock them down when their flaws might result in security issues.

Linus replied, “All of this is pure garbage. Is Intel really planning on making this shit architectural? Has anybody talked to them and told them they are f\*cking insane? Please, any Intel engineers here – talk to your managers.”

David Woodhouse said, “If the alternative was a two-decade product recall and giving everyone free CPUs, I’m not sure

it was entirely insane.” He agreed that KarimAllah’s patch was a “nasty hack,” but added, “hey – the world was on fire, and in the end we didn’t have to just turn the datacentres off and go back to goat farming, so it’s not all bad.”

Linus replied, “You seem to have bought into the cool-aid. Please add a healthy dose of critical thinking. Because this isn’t the kind of cool-aid that makes for a fun trip with pretty pictures. This is the kind that melts your brain.”

He went on to say that the true situation was much worse than a “nasty hack.”

First, Linus acknowledged that, “The speculation control cpuid stuff shows that Intel actually seems to plan on doing the right thing for Meltdown (the main question being `_when_`). Which is not a huge surprise, since it should be easy to fix, and it’s a really honking big hole to drive through. Not doing the right thing for Meltdown would be completely unacceptable.”

But he went on to say, “the IBRS garbage implies that Intel is `_not_` planning on doing the right thing for the indirect branch speculation. Honestly, that’s completely unacceptable too.”

He continued:

*The whole IBRS\_ALL feature to me very clearly says “Intel is not serious about this; we’ll have a ugly hack that will be so expensive that we don’t want to enable it by default, because that would look bad in benchmarks.”*

*So instead they try to push the garbage down to us. And they are doing it entirely wrong, even from a technical standpoint.*

*I’m sure there is some lawyer there who says “we’ll have to go through motions to protect against a lawsuit.” But legal reasons do not make for good technology, or good patches that I should apply.*

Earlier, David had remarked, “We do need the IBPB feature to complete the protection that retpoline gives us – it’s that or rebuild all of userspace with retpoline.” To which Linus said:

**BULLSHIT.**

*Have you `_looked_` at the patches you are talking about? You should have – several of them bear your name.*

*The patches do things like add the garbage MSR writes to the kernel entry/exit points. That’s insane. That says “we’re*

*trying to protect the kernel.” We already have retpoline there, with less overhead.*

*So somebody isn’t telling the truth here. Somebody is pushing complete garbage for unclear reasons. Sorry for having to point that out.*

*If this was about flushing the BTB at actual context switches between different users, I’d believe you. But that’s not at all what the patches do.*

*As it is, the patches are COMPLETE AND UTTER GARBAGE.*

*They do literally insane things. They do things that do not make sense. That makes all your arguments questionable and suspicious. The patches do things that are not sane.*

**WHAT THE F\*CK IS GOING ON?**

*And that’s actually ignoring the much `_worse_` issue, namely that the whole hardware interface is literally mis-designed by morons.*

*It’s mis-designed for two major reasons:*

- the “the interface implies Intel will never fix it” reason.
- the “there is no performance indicator” [reason]

*The whole point of having cpuid and flags from the microarchitecture is that we can use those to make decisions. But since we already know that the IBRS overhead is huge on existing hardware, all those hardware capability bits are just complete and utter garbage. Nobody sane will use them, since the cost is too damn high. So you end up having to look at “which CPU stepping is this” anyway. I think we need something better than this garbage.*

David was actually in full agreement. Instead of Intel making a fix “opt-in,” he said, they should just fix it, since failing to opt in would always be unacceptable. But he felt that for the kernel, a short-term fix that was horrendously ugly would still be OK, given that it was just a short-term fix.

He also pointed out that Linus had been looking slightly in the wrong place when complaining about David’s patches – David pointed out that Linus had been looking at IBRS rather than IBPB. To which Linus replied, “Ehh. Odd Intel naming detail.” And he went on, “If

you look at this series, it very much does that kernel entry/exit stuff. It was patch [number 10 of 10 in the series]. In fact, the patch I was replying to was explicitly setting that garbage up. And I really don’t want to see these garbage patches just mindlessly sent around.”

David replied with a very interesting and useful summary of the whole situation. He said:

*I think we’ve covered the technical part of this now, not that you like it – not that any of us *\*like\** it. But since the peanut gallery is paying lots of attention, it’s probably worth explaining it a little more for their benefit.*

*This is all about Spectre variant 2, where the CPU can be tricked into mispredicting the target of an indirect branch. And I’m specifically looking at what we can do on *\*current\** hardware, where we’re limited to the hacks they can manage to add in the microcode.*

*The new microcode from Intel and AMD adds three new features.*

*One new feature (IBPB) is a complete barrier for branch prediction. After frobbing this, no branch targets learned earlier are going to be used. It’s kind of expensive (order of magnitude ~4,000 cycles).*

*The second (STIBP) protects a hyper-thread sibling from following branch predictions which were learned on another sibling. You *\*might\** want this when running unrelated processes in userspace, for example. Or different VM guests running on HT siblings.*

*The third feature (IBRS) is more complicated. It’s designed to be set when you enter a more privileged execution mode (i.e., the kernel). It prevents branch targets learned in a less-privileged execution mode, BEFORE IT WAS MOST RECENTLY SET, from taking effect. But it’s not just a “set-and-forget” feature, it also has barrier-like semantics and needs to be set on*



*\*each\* entry into the kernel (from userspace or a VM guest). It's *\*also\** expensive. And a vile hack, but for a while it was the only option we had.*

*Even with IBRS, the CPU cannot tell the difference between different userspace processes, and between different VM guests. So in addition to IBRS to protect the kernel, we need the full IBPB barrier on context switch and vmexit. And maybe STIBP while they're running.*

*Then along came Paul with the cunning plan of "oh, indirect branches can be exploited? Screw it, let's not have any of *\*those\** then," which is *retpoline*. And it's a *\*lot\** faster than frobbing IBRS on every entry into the kernel. It's a massive performance win.*

*So now we *\*mostly\** don't need IBRS. We build with *retpoline*, use IBPB on context switches/vmexit (which is in the first part of this patch series before IBRS is added), and we're safe. We even refactored the patch series to put *retpoline* first.*

*But wait, why did I say "mostly"? Well, not everyone has a *retpoline* compiler yet ... but OK, screw them; they need to update.*

*Then there's Skylake, and that generation of CPU cores. For complicated reasons they actually end up being vulnerable not just on indirect branches, but also on a "ret" in some circumstances (such as 16+ CALLs in a deep chain).*

*The IBRS solution, ugly though it is, did address that. *Retpoline* doesn't. There are patches being floated to detect and prevent deep stacks and deal with some of the other special cases that bite on SKL, but those are icky too. And in fact IBRS performance isn't anywhere near as bad on this generation of CPUs as it is on earlier CPUs *\*anyway\**, which makes it not quite so insane to *\*contemplate\** using it as Intel proposed.*

*That's why my initial idea, as implemented in this RFC patchset, was to stick with IBRS on Skylake and use *retpoline* everywhere else. I'll give you "garbage patches," but they weren't being "just mindlessly sent around." If we're going to drop IBRS support and accept the caveats, then let's do it as a conscious decision having seen what it would look like, not just drop it quietly because poor Davey is too scared that Linus might shout at him again. :)*

*I have seen *\*hand-wavy\** analyses of the Skylake thing that mean I'm not ac-*

*tually lying awake at night fretting about it, but nothing concrete that really says it's OK.*

*If you view *retpoline* as a performance optimization, which is how it first arrived, then it's rather unconventional to say "well, it only opens a *\*little\** bit of a security hole, but it does go nice and fast, so let's do it."*

*But fine, I'm content with ditching the use of IBRS to protect the kernel, and I'm not even surprised. There's a *\*reason\** we put it last in the series, as both the most contentious and most dispensable part. I'd be *\*happier\** with a coherent analysis showing Skylake is still OK, but hey-ho, screw Skylake.*

*The early part of the series adds the new feature bits and detects when it can turn KPTI off on non-Meltdown-vulnerable Intel CPUs and also supports the IBPB barrier that we need to make *retpoline* complete. That much I think we definitely *\*do\** want. There have been a bunch of us working on this behind the scenes; one of us will probably post that bit in the next day or so.*

*I think we also want to expose IBRS to VM guests, even if we don't use it ourselves. Because Windows guests (and RHEL guests; yay!) do use it.*

*If we can be done with the shouty part, I'd actually quite like to have a sensible discussion about when, if ever, we do IBPB on context switch (ptraceability and dumpable have both been suggested) and when, if ever, we set STIBP in userspace.*

Luke Kenneth Casson Leighton thanked David for this explanation, saying, "there is actually a significant benefit to what you're doing, not just peanut-gallery-ing." He said, "this is a cluster-f\* where every single Intel (and AMD) engineer is prevented and prohibited from talking directly to you as they develop the microcode. ... They've been ignored and demoralized. It's a lesson that I'm not sure their management are capable of learning, despite the head of the Intel open source innovation centre has been trying to get across to them for many years: OPEN UP THE FUCKING FIRMWARE AND MICROCODE."

He went on, "unfortunately, the burden is on you, the members of the linux kernel team, to read between the lines, express things clearly here on LKML so that the intel engineers who are NOT

PERMITTED to talk directly to you can at least get some clear feedback ... to indicate *\*to them\** that you fully grasp the technical situation ... whilst at the same time not being permitted access to the fucking microcode."

Meanwhile, David was joined by Ingo Molnár and many others working on the technical side of generating new patches to solve the various issues.

There was clearly a lot of bitterness among the kernel developers – for example, as expressed by Pavel Machek at one point in the discussion, where he said, "someone at Intel put world on fire. And then was selling faulty CPUs for half a year while world was on fire; they knew they are faulty yet they sold them anyway. Then Intel talks about how great they are and how security is important for them .... Intentionally confusing between Meltdown and Spectre so they can mask how badly they screwed. And without apologies."

He went on to say, "and now Intel wants to cheat at benchmarks, to put companies that do right thing at disadvantage and thinks that that's okay because world was on fire? At this point, I believe that yes, product recall would be appropriate. If Intel is not willing to do it on their own, well, perhaps courts can force them. Ouch and I would not mind some jail time for whoever is responsible for selling known-faulty CPUs to the public."

The discussion continued for quite awhile, mostly on a technical level.

Aside from unhappiness toward Intel, the issue is fascinating and helps illuminate some of the fundamental issues confronting a universe in which open source has already essentially successfully taken over the world. How can the relationships and conflicting motivations between open source projects and for-profit corporations be properly managed? How can private companies reveal crucial information about proprietary microcode and such things, while protecting their competitive secrets from other corporations? And if, as Linus guessed in an early post, Intel intends to maintain a significant flaw over the long term, how can open source developers and users communicate with, and perhaps put pressure on, such companies to change that kind of decision? ■■■



## Transparency in firmware with Libreboot

# Liberated

Libreboot is a project dedicated to making firmware free as in freedom. *By Erik Bärwaldt*

**F**or users who believe in free software, one of the biggest benefits is that, because you have access to the source code, you know what your computer is doing. Proprietary, closed-source systems contain binary-only code that might be compromising your security and privacy – and you wouldn't even know it.

Linux and other open source systems offer a level of transparency that simply isn't possible with proprietary systems; however, even if you are using an all-free distro recommended by the Free Software Foundation (FSF) [1], your system might not be as free as you think. Most computers today come with proprietary firmware that boots the system and manages certain low-level tasks related to the hardware. Proprietary firmware has many of the same problems proprietary software has: You don't really know what it is doing, and it could be putting your security and privacy at risk.

BIOS, and its successor UEFI, are standards for the firmware systems preinstalled into most of the personal computers sold around the world, and they are generally *not* distributed under a free license. In addition, Intel's management engine, which has been integrated into hardware since 2008, is a thorn in the side of the proponents of completely free systems, allowing remote access and opening up several possibilities for attack. Another problem with proprietary firmware for users with high security needs is microcode updates, which are supposed to improve the microcode of the CPU but add an extra layer of uncertainty and potential vulnerability.

For all these reasons, resourceful developers have begun to replace the remaining non-free components in computer systems with free alternatives – including free firmware.

Coreboot [2] is a community project dedicated to developing an open source firmware alternative. Coreboot, which was formerly known as the LinuxBIOS project, claims to support over 230 main boards. In addition to bringing transparency to the firmware, Coreboot also claims to improve boot time by removing some of the unnecessary features and code bloat associated with mainstream firmware systems.

## Players

Several small international companies have emerged around free BIOS implementations. These vendors fill the niche of providing computers for users who want a system that is truly all free. They can also offer the benefit of faster boot time with a trimmed down system, and they are in a unique position to avoid the planned obsolescence of the computer industry by maintaining support for older systems that are no longer relevant to the big hardware vendors.

Nearly all of these companies primarily offer Lenovo laptops. The economies of the free firmware industry mean that the companies tend to work with large volumes of older computers: the most common models are series X200, T400, and





T500 ThinkPads, which the original manufacturer produced between 2008 and mid-2010.

These systems are based on the Core 2 duo processors of Intel's Penryn architecture, which have two cores but lack performance-enhancing features such as turboboost and hyper-threading.

The performance of these devices is fine for office and Internet applications, but demanding tasks such as CAD applications or video transcoding push the hardware to its performance limit. Many modified Lenovo notebooks are certified by the FSF.

(The US provider Purism is an exception to this low-end-hardware strategy. See the box entitled "Purism: A Different Approach.")

### Libreboot

The Libreboot project [3] was founded in December 2013 in the UK with the goal of replacing the BIOS and video BIOS of common GPUs with free components by reverse engineering. Libreboot, which is written in C and Assembler, is a derivative of coreboot, but the Libreboot developers remove some remaining proprietary firmware blobs that still exist in the coreboot. Libreboot programmers do not try to compete with coreboot but participate actively in its development by contributing patches and simplified installation processes.

Libreboot was simultaneously implemented on several server and desktop systems with standard components, as well as on some notebooks. Since extensive adjustments for the chipset and motherboard were necessary, the software runs on a limited number of hardware components. The worldwide distribution of the devices is managed by a company called Minifree [4]. The name *Minifree* is short for Ministry of Freedom – a reference to the novel *1984*.

Minifree was founded in December 2014 by 23-year-old Leah Rowe (see the "Meet Libreboot and Minifree Founder

Leah Rowe" box). The company only sells computers that have been upgraded with Libreboot. Minifree contributes to the financing of the coreboot distribution, and it also pays for additional developers who port Libreboot to new hardware.

Depending on availability of the necessary hardware, Minifree offers a freely configurable server and desktop system, notebooks, and accessories. The devices are mostly systems from older processor generations, because the great effort for the implementation of Libreboot allows only limited modifications, especially for laptops.

In addition to the actual BIOS, the developers also adapt the embedded controllers, removing Intel's management engine or the equivalent Secure Processor from AMD. In addition, they replace problematic network hardware with components that do not require proprietary firmware, such as Qualcomm/Atheros components.

After the upgrade, the computer systems are completely free of closed source software, which means a distribution that does not use proprietary binary blobs and does not require separate firmware for the operation of some hardware components runs without any functional impairment.

The Lenovo ThinkPad T400 notebooks, currently offered by Minifree and certified by the FSF under the "Respects Your Freedom" program, are therefore completely free systems. These systems come with a two-year guarantee at prices of between just under EUR240 to EUR630 (~\$296-\$777), plus shipping.

### Operating System

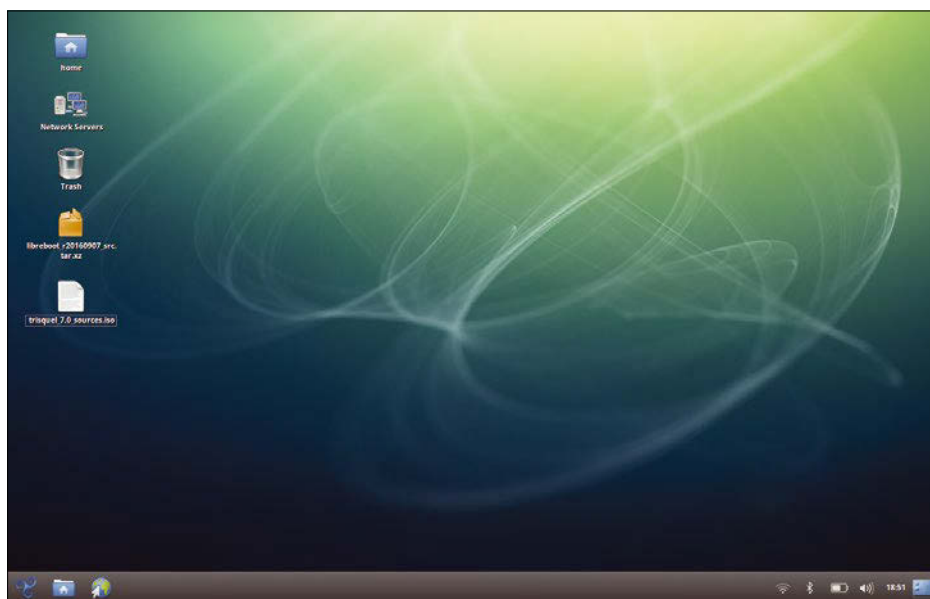
Minifree preinstalls Trisquel Linux on its devices. Trisquel is an Ubuntu derivative from which the developers have removed any non-free components [5]. The system also differs visually from the baseline Ubuntu system: Instead of the fuzzy brown and purple tones in the original, Trisquel features fresh blue and green, with application windows featuring 3D effects in light gray shades (Figure 1).

Trisquel Linux, which originates in Spain, has the approval of the FSF as a completely free operating system. Minifree provides the complete source code for Trisquel, totaling almost 5GB, as well as the source code for Libreboot, which weighs in at several hundred megabytes.

### Purism: A Different Approach

US provider Purism is a laptop vendor that specializes in free firmware, but Purism doesn't depend on older systems. The company is itself a laptop manufacturer that uses state-of-art hardware. Purism equips the devices with coreboot and removes other closed-source elements: For example, the company's Libre OS notebooks exclusively use WLAN cards with the free-firmware-friendly Atheros chipsets. In addition to the coreboot BIOS, Purism uses the Debian-derived PureOS and Qubes OS operating systems.

The capitalization of the company demonstrates the value of this business idea: Purism received approximately US\$2.5 million in working capital as part of a crowdfunding scheme. This influx of capital enabled the company to modify its original concept of built-to-order and now sell their notebooks from stock – eliminating months of delivery time.



**Figure 1:** The Trisquel Linux desktop looks fresh and friendly.

### Test Device

We ordered a Lenovo ThinkPad T400 from Minifree with Trisquel and Libreboot preinstalled. Just three days after the order, an unexpectedly large package arrived with a 14-inch notebook (Figure 2). In addition to the ThinkPad and its power supply unit, the package included a patch cable and two power cords, one with a plug for continental Europe and the other with the British three-pin connector and a built-in fuse.

The ThinkPad T400 has a 14-inch display in the 16:10 form factor with a resolution of 1280x800 pixels. The Core 2 duo processor of the “Penryn” generation includes energy-saving features, running at a maximum clock speed of 2.4GHz. The T400 comes with 4GB of RAM and a 160GB conventional SATA hard disk. The rechargeable battery supports mobile use for up to six hours.

Although the test notebook was created in May 2009 and is therefore more than eight years old, it looks almost as good as new. The positive impression of the hardware was confirmed by reading out the parameters of the still-sealed hard disk: The test system was in use for just under 7100 hours of operation – an unusually low number for a computer of this age. The original keyboard, as well as the case itself, show no discernible signs of usage even if you look closely. Minifree also includes a two-page quick start guide that explains the authentication data provided in Trisquel Linux.

In addition to the free BIOS and the VBIOS modification, the device has another special feature: An Atheros WLAN card that supports free firm-

ware replaces the original Intel WLAN card.

After the device switches on, the rapid speed is surprising. The Libreboot BIOS, which does not display a ThinkPad logo, works significantly faster than the Lenovo BIOS. Trisquel Linux also boots considerably faster than other systems on similar hardware.

### Unusual

Unlike conventional BIOS or UEFI systems, Libreboot does not offer a way to call a dialog from the firmware at startup that lets you configure the boot order or other settings. Instead, a GRUB selection menu opens directly. You can choose the medium from which the computer will boot.

It is possible to select the desired option using a hotkey. The most important functions are *[O]* to start the installed Trisquel with an encrypted hard disk, *[U]* to boot from a storage medium connected via the USB bus, *[D]* to boot the system from an optical disc, and *[S]* to find a GRUB 2 configuration on an external disk. You will also find a *Poweroff[p]* option for shutting down the system and a *Reboot[r]* option for restarting the system.

In our lab, manually burned CDs and DVDs from ISO images functioned as boot media without any problems. In other cases, the boot options failed: For example, distributions preinstalled



**Figure 2:** The T400 from Minifree arrived in our test lab in practically new condition.



### Listing 1: Transferring with dd

```
$ dd if=/image_name of=/dev/device_name bs=4M; sync
```

on different USB media would not start, although they started on similar Fujitsu-Siemens and Hewlett-Packard devices with a non-free BIOS. *Linux Magazine* DVDs generally wouldn't boot.

The Libreboot project describes options for manually booting from external discs in its detailed documentation. The developers recommend transferring ISO images to data media such as USB memory using `dd` (Listing 1). Accordingly, we prepared several USB sticks of different types with different distributions. All manually created flash memory devices booted without any problems. Media created with tools like UNetbootin or the ROSA Image Writer did not work.

When booting from removable media, the startup time was significantly shorter compared with conventional computers. We experienced no problems with hardware components during the test mode. Even the modified VBIOS of the ThinkPad T400 caused no problems (Figure 3).

### Weak Points

As expected, the “liberated” ThinkPad T400, in combination with Trisquel Linux, displays weaknesses where developers have not yet replaced proprietary blobs and firmware with free components. This applies in particular to WWAN devices that could not be talked into cooperating, such as CardBus plugin cards. In view of the thousands of components and devices available on the market, it is hardly surprising that a small project like Libreboot is not able to equip all these components with completely free firmware.

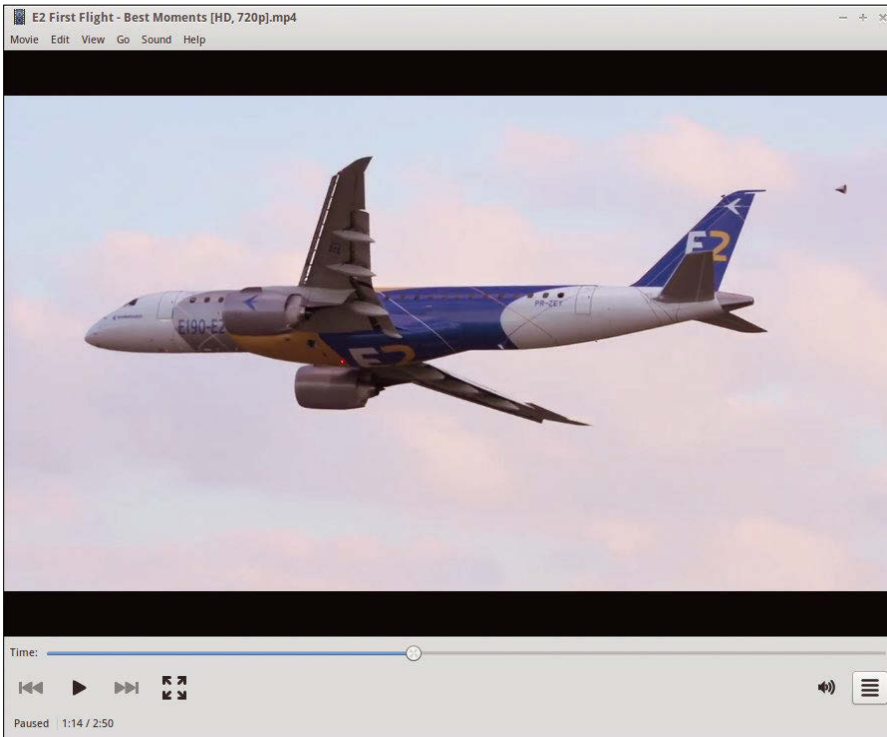


Figure 3: Even movies work on Trisquel Linux without any problems.

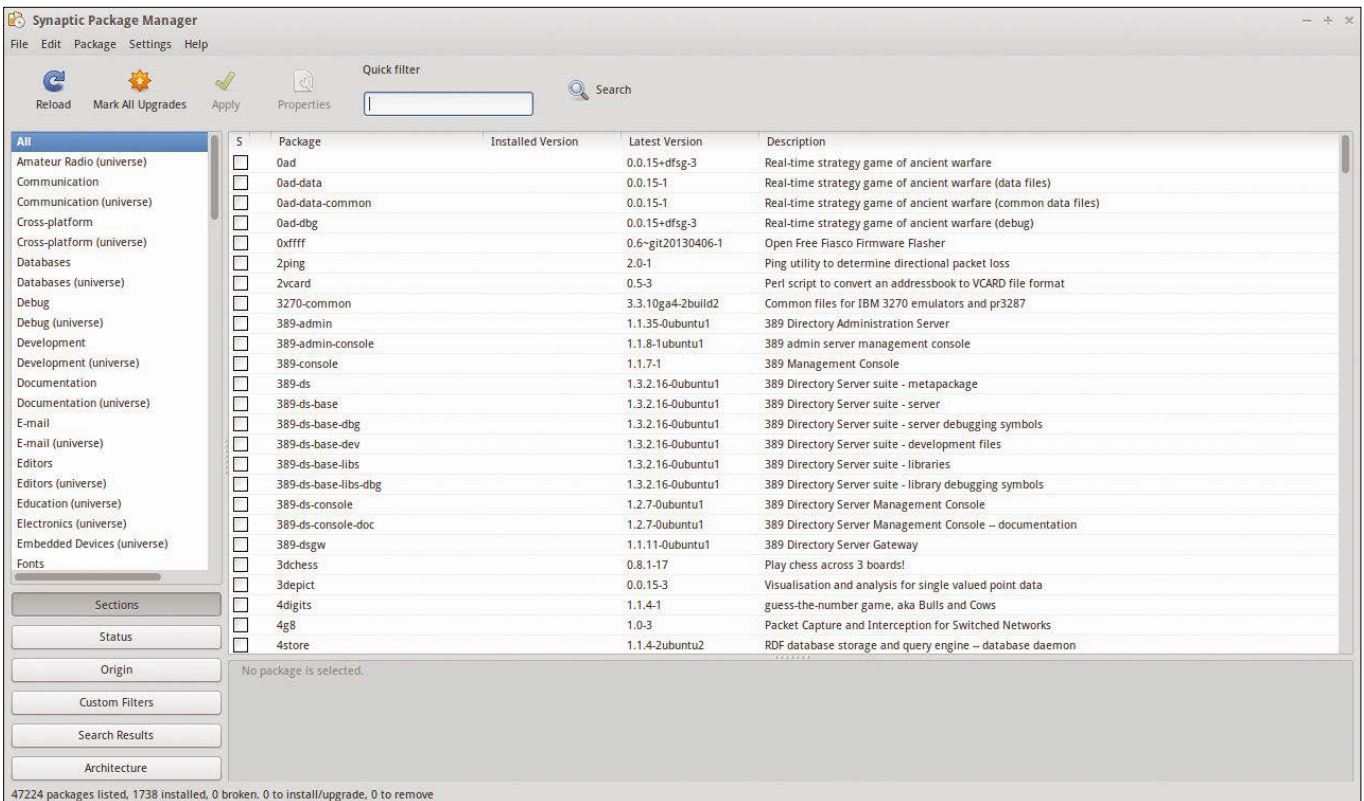


Figure 4: Thanks to Ubuntu's software repositories, almost all users get their money's worth.



Non-free software codecs, which are completely missing in Trisquel Linux but which are necessary for playing certain multimedia content, also limit the possible applications. Nevertheless, the complete system with more than 47,000 packages from the Ubuntu repositories is completely suitable for everyday use (Figure 4).

### Conclusions

Free software again demonstrates its value when it replaces hardware-specific proprietary firmware: Libreboot even gives older equipment a boost and has a positive influence on the behavior of the system. The ThinkPad T400 systems from Minifree are ideal for tough everyday uses thanks to their careful programming and configuration. Let's hope that the development team led by Leah Rowe will receive more support for porting Libreboot to new hardware. ■■■

### Info

- [1] FSF Free GNU/Linux Distributions: <https://www.gnu.org/distros/free-distros.en.html>
- [2] coreboot: <https://www.coreboot.org/>
- [3] Libreboot: <https://libreboot.org/>
- [4] Minifree: <https://minifree.org>
- [5] Trisquel Linux website: <https://trisquel.info/>
- [6] AMD-PSP: <https://www.amd.com/en/technologies/security>
- [7] RISC-V project: <https://riscv.org>

### Meet Libreboot and Minifree Founder Leah Rowe

**LM:** *Minifree Ltd. currently only offers a laptop model, a modified Lenovo ThinkPad T400. [Note: The ThinkPad X220 also became available since the time of this interview.] Although technically and visually in very good condition, these machines are around eight years old. Lenovo has long been distributing more modern computers with considerably higher performance. Libreboot is not yet available for more modern devices with the Core i architecture. Is there any chance that this will change soon?*

**Leah Rowe:** We're working on setting up Libreboot for Sandy Bridge systems. Specifically, we are porting to ThinkPad X220 laptops. After that, we'll work on the ThinkPad T420.

This work is still in progress because we still need the non-free firmware for Intel's management engine to operate the devices. Without this firmware, the system performs a hard reset every 30 minutes – we're trying to eliminate this. In the past, there was the ThinkPad X200 with Libreboot; we want to offer this option again in the future. In addition, we want to accelerate the sale of the T400 so that we have more resources to finance the X220 ports.

**LM:** *Is there any support from well-known hardware manufacturers for the Libreboot project? Have you tried to get support?*

**LR:** No. We do not receive any support from the manufacturers. We have tried to convince AMD to support us. At the beginning of 2017, we talked to AMD about cooperating with coreboot and Libreboot, which would have led to the release of non-free PSP firmware [6]. I was just told that my request had been forwarded to the responsible team.

I no longer believe that these manufacturers will ever cooperate with us. We are more oriented towards the RISC-V project [7], which develops a completely free Instruction Set Architecture for a CPU.

RISC-V is not yet usable in laptops, workstations, or servers, but it is already suitable for developer boards or for a microcontroller setup. With RISC-V, it would be possible in the future to offer large-scale OEM systems with completely free hardware, but that will take a few more years.

**LM:** *What support is Libreboot currently receiving from the free software community? And what support do you want – apart from programmers who can help develop free firmware?*

**LR:** Libreboot has recently appointed a new full-time maintainer for the project. On our website, we offer an overview for developers who wish to contribute. Recently, a number of people who have not been involved in the project have contributed to Libreboot.

My main concern is to get more support outside programming. Anyone can participate in the Libreboot project – even without technical experience: For example, volunteers can provide help to users on the IRC channel or help improve the documentation.

There is also a Librecore project that Libreboot is working with. Many highly qualified developers work for Librecore, setting up new hardware ports and making them available for Libreboot. Libreboot is currently based on coreboot, but in the future, we want to use Librecore as an upstream for hardware initialization.

Librecore is a fork of coreboot. Today, Libreboot is a coreBoot distribution; in the future, it will be a Librecore distribution. Primarily, we want to attract more supporters, so that their work will allow us to use more hardware and Libreboot's list of supported hardware will continue to grow.



**GOT CLUSTER?**  
Tune in to the HPC Update newsletter for news, views, and real-world technical articles on high-performance computing.

**HPC Up Close**

- Let the Editor Wars Begin!
- Los Alamos Double-Checks Wiring: Saves \$2 Million
- XSEDE Starts a New Era
- TKperf

**Highlights**

**Let the Editor Wars Begin!**  
Editors, particularly command-line editors, are an important tool for high-performance computing systems administrators. We ports out some...

**XSEDE Starts a New Era**  
NSF announces details for new XSEDE 2.0 phase

**Los Alamos Double-Checks Wiring: Saves \$2 Million**  
A review team for the Trinity supercomputer finds savings with aluminum cable and other innovations.

**TKperf**  
Linux offers several low-level tools for investigating solid state drives (SSDs). Admins are well advised to deploy these tools immediately...

**Most Read**

**SSD Tuning**  
Solid-State drives (SSDs) are quite different from old-fashioned hard disks, and you'll have to learn some new techniques if you want to tune up...

**Boosting Performance with Intel's QuickAssist Technology**  
Quickassist technology offloads computationally intensive compression and encryption tasks to provide a performance boost for intel processors.

**Workshop on Workflows Announced**  
Free training session for the HPC community will take place at 12 US locations

**Further Reading**

- Useful NFS Options for Tuning and Management
- New Service Will Adopt HPC Code for Next-Generation Hardware
- Singularity – A Container for HPC
- Performance Management Tools
- Desired State Configuration for Linux

**Altair Case Study**

Watch the Wayne State University video on using PBS Pro to support their environment for research.

**Get to Know ADMIN Magazine**

Get to know ADMIN with a risk-free ADMIN magazine Trial Subscription.

Try ADMIN magazine for two issues delivered to your door or your inbox over four months with the No-Risk Trial Subscription.

You'll be sent the next two issues of ADMIN magazine for the cost of postage.

[hpc.admin-magazine.com/newsletter](http://hpc.admin-magazine.com/newsletter)



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

## Expand your Linux skills with:

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

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

**Subscribe now!**  
[shop.linuxnewmedia.com/subs](http://shop.linuxnewmedia.com/subs)

**GET IT NOW!**  
 FAST DELIVERY  
 WITH OUR PDF EDITION



## Collaborative workflows

# Shared Office

Several collaborative office suite providers are now competing for customers who are looking for the seamless exchange of office documents within an enterprise. These cloud or on-premise solutions are designed for teamwork. *By Erik Bärwaldt*

**T**oday, shifting documents from employee to employee by email attachment is considered “so 90s.” No wonder: Office suites enable completely new forms of cooperative work, often via cloud or on-site servers. This article takes a look at the rather confusing market that has developed around collaborative office suites.

### Concepts

There is no uniform approach to what collaborative work looks like. Different methods reflect different visions of the modern work environment. However, one important distinction is between a client/server-based collaborative working environment and web-based collaboration.

Client/server structures store all data processed by a team on a server. A client application uses the Message Application Programming Interface [1] to

allow simultaneous use of the data. Typical applications include diary, address management, and project management software.

The other approach to collaborative software involves web-based solutions storing data and software packages centrally on a self-managed server. Users access the service via a web browser. On the upside, this concept offers lower administrative overhead: You do not have to install and maintain any applications on the workstations, and a web browser is standard on modern computer systems. The installation’s potentially inferior reliability is the downside: If the server fails, the whole system ceases to function.

In addition, there are also purely cloud-based programs in which both the applications and the data storage reside in the cloud. The advantage: Installing

servers on site often proves to be complicated and error-prone. Users leverage cloud as Software as a Service (SaaS) [2] to avoid the need to install a server variant onsite – operators completely manage the services centrally.

### Uncharted Territory

When it comes to editing texts, tables, and presentations, groupware-capable solutions do not yet play as important a role as in traditional collaborative tasks: Project management, PIM, or CRM systems have been focusing on cooperation for some time now – they no longer exist as single-user versions.

Numerous Linux providers lag behind in terms of office teamwork solutions. They deliver office suites that are suitable for use in workgroups. However, there is a sticking point: If users work in environments with a range of operating

## Noteworthy

In addition to traditional groupware office suites, which cover the entire spectrum of office work with texts, tables, and presentations, Linux supports several small projects that solve sub-tasks.

Particularly noteworthy are EtherPad Lite [3] and EtherCalc [4]. They are dedicated to collaborative work with texts and tables. Both projects work across platforms and are web-based. You store the individual texts or tables in “pads” on a web server. Every user who knows the URL can contribute to the text. These programs color-highlight contributions by the different users. Thus, even if several users work simultaneously on a text or a table, you can still keep track.

Celtx [5] allows collaborative work on screenplays, focusing on media produc-

tion. It is available in a desktop version, but effective use requires a commercial account with the software developer. The scripts can be stored in a cloud-based, encrypted form on the manufacturer’s servers. In this way, users can also make the scripts accessible to others.

Prezi [6], online presentation software based on HTML5, is quite popular with users. It provides a virtual whiteboard and allows access for up to 10 users. Prezi is available for a subscription fee and does not offer native Linux applications for local editing or presentation playback. An alternative is Sozi [7], a free plugin for Inkscape, which is functionally very similar to Prezi. However, unlike older Prezi variants, Sozi does not require Adobe’s Flash technology, but instead generates SVG graphics.

systems, they need to easily exchange files without any problems; they don’t want to worry about converting between file formats.

## Collabora Online

Collabora Online (Figure 1) might be familiar to users: The Collabora Office [8] enterprise desktop solution from the United Kingdom is a customized LibreOffice that has been modified to work in the cloud on a server.

Collabora Online benefits from long-term support and is available across platforms. The manufacturer provides commercial support services via various service partners worldwide. Beyond this, Collabora Online Development Edition (CODE) is another web-based teamwork solution derived from Collabora Office core. Ultimately, both variants allow users to create and edit documents on web browser clients. This works without requiring users to install

the office suite locally on their desktop systems.

However, in contrast to Collabora Online, CODE is rather experimental in character. The company, therefore, does not recommend it for production use. It is available as a Docker image for use in the home office and is more likely to be used for testing purposes [9]. If you want to use the two packages, you need an ownCloud, Nextcloud, Pydio, or Seaf- file infrastructure as a prerequisite.

As a third package, the company distributes the Collabora CloudSuite (see [10] for a demo). CloudSuite is a combination of an online and a desktop editor. It includes a local installation of Collabora Office as well as the cloud variant. The result is that CloudSuite makes documents available at the workplace and simultaneously pushes them onto smartphones or tablets.

## Offline

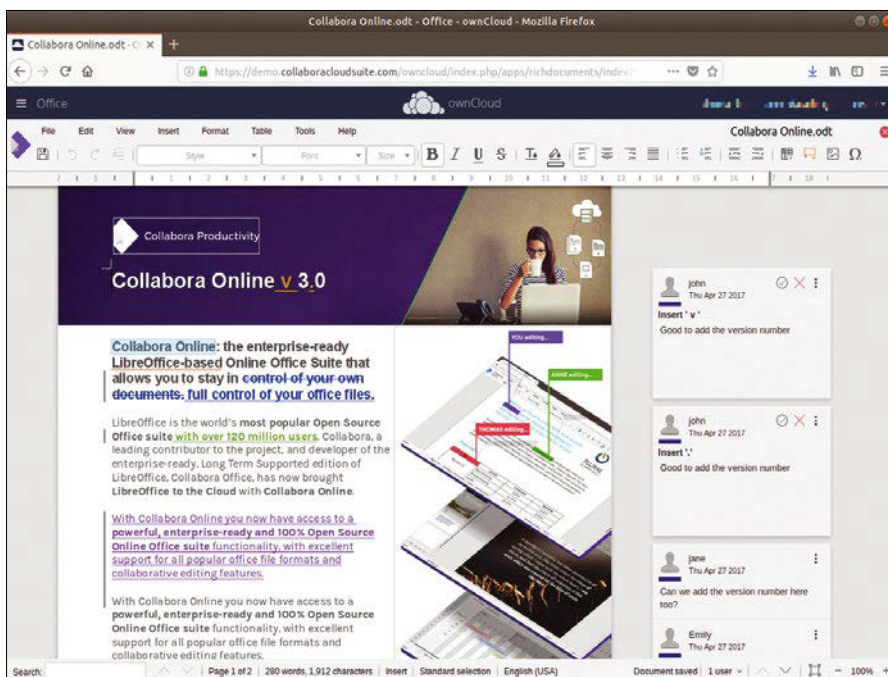
Collabora also offers a complete office suite based on LibreOffice for users who don’t have a server. Compared to the free original, it primarily stands out with conversion filter improvements (i.e., it converts documents into different formats in a superior way). Currently, version 5.1 is available in a 680MB Debian package for 64-bit hardware. The LibreOffice version also includes additional language packs and help texts.

## Functional Overkill

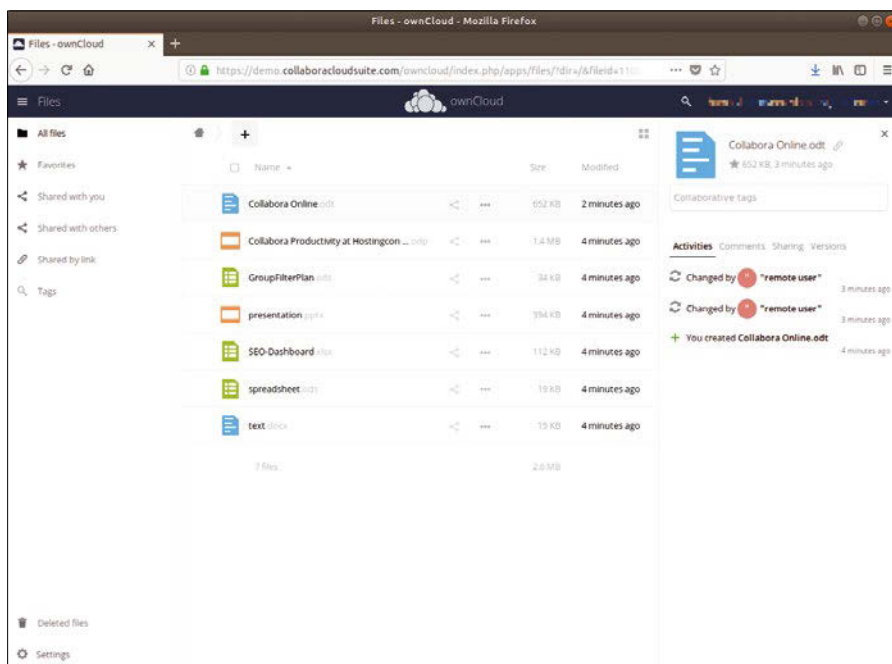
Compared to a locally installed office package, the online versions offer a significantly reduced range of functions in all areas. The editing options are similar to those of a LibreOffice, but you have to considerably restrict the formatting options.

The online versions only accept DIN A4 and A5 paper formats, as well as the US letter and legal sizes. Even the formatting of inserted tables quickly reaches its limits. In terms of import and export formats, some of the less common file formats are missing. You can only download documents to your local system in four formats, the software distinguishes between text, table, and presentation formats.

A positive feature is the comment function, which is important for collaborative work. Users can access this in all documents via *Paste | Comment*.



**Figure 1:** Collabora Online is a functionally streamlined LibreOffice with teamwork components.



**Figure 2:** In the right-hand pane, you can see all the important data on a document's processing status.

In a separate window, you then enter a short comment, which appears to the right of the actual document after clicking *Save* in the main window. The various team members' comments all appear together. This makes it easy to assign annotations to the document text, similar to the comment function in offline Office packages.

Users can track changes in the documents without difficulty. For each application, you can switch a logging function on and off via *Edit | Track changes*. This feature also allows you to scroll between the individual changes. This gives the user a quick overview of document versions.

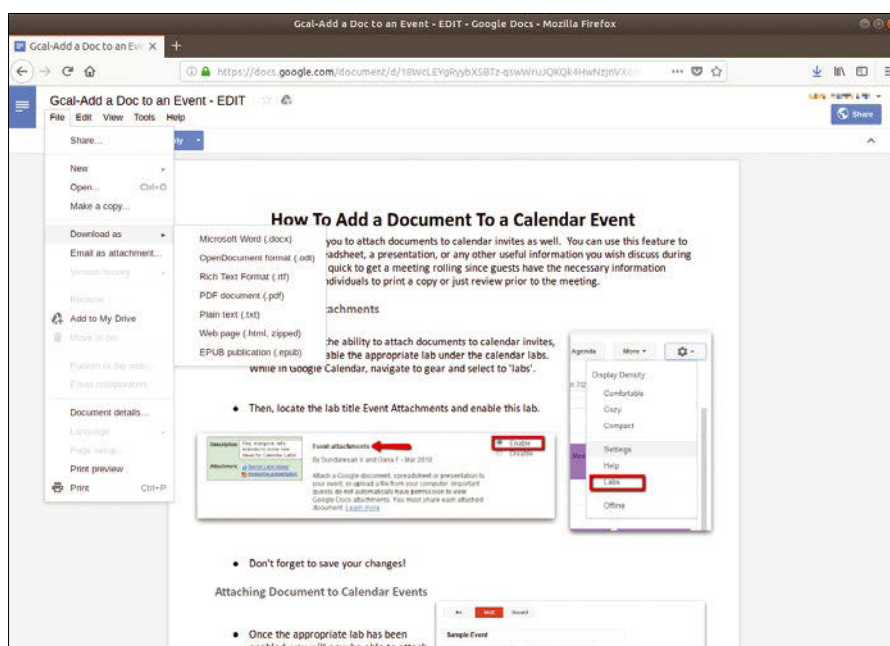
At the same time, Collabora Online presents these functions in the list view on the right-hand side of the screen; a mouse click opens an area. In this way, you can immediately see all the relevant data on the current document's processing status (Figure 2) without having to click your way through complex menu hierarchies.

### Google Docs

Search engine giant Google interweaves its online office package Google Docs [11] tightly with the company's many other services. The package itself consists of four components: Google Docs, Google Spreadsheets, Google Presentations, and Google

Forms. The Google suite is completely cloud-based. Consequently, local desktop computer clients don't exist, and on-premise installation on local dedicated cloud servers is not possible. Google only offers apps for smartphones and tablets; on stationary systems, the software runs in the web browser.

To work with Google Docs (Figure 3), you need a Google account. Once you have set up an account, you can log on to the office packages separately.



**Figure 3:** Google Docs is completely web-based; clients are only available for smartphones and tablets.

### Functions

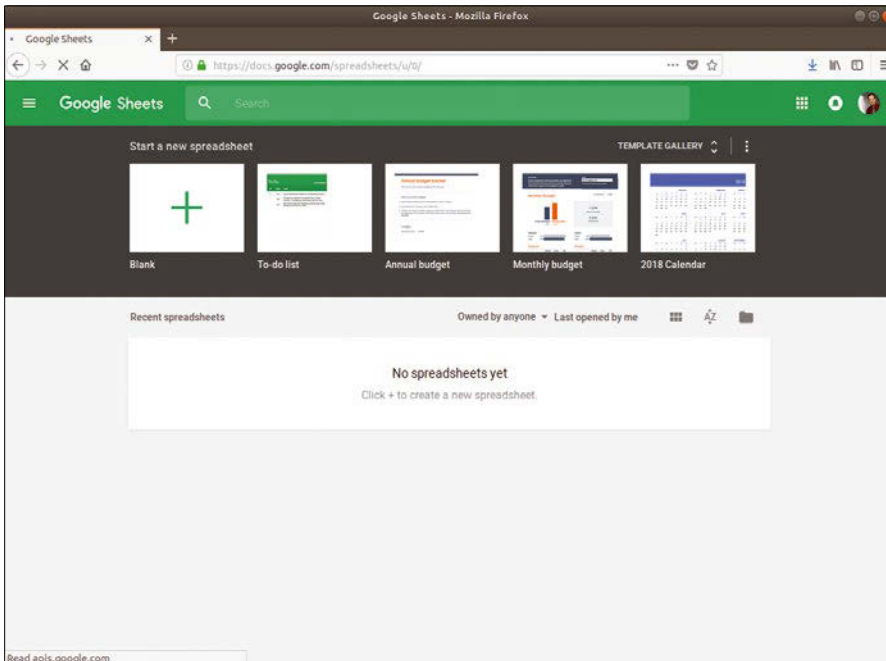
Google Docs provides all the features needed for collaborative work. This includes basic document editing, comments, and file sharing. If desired, two users can edit a document simultaneously. In this case, one participant can see almost in real time the changes the other participant is making based on the cursor movements.

Another unique selling point is the modular concept, which means you can extend the applications with add-ons such as dictionaries and templates. One disadvantage of the add-on architecture is that Google tightly links Docs to its own web browser; many add-ons therefore only run in Chrome.

Google Docs does not have the ability to save documents manually. Instead, the provider automatically saves them in the Google Cloud. However, downloading and converting the data is relatively flexible: You can download your files in different formats to your local system, where you define the desired formats via *File | Download as*. The individual options are configurable in a context-sensitive approach.

You will also notice that older Microsoft formats are missing in the export function. In addition, Google Docs offers an unusual user interface: You do not see an empty document at the beginning; instead, to open a new file, you have to explicitly click on the red plus





**Figure 4:** When a new file is created, the user interface is initially empty.

symbol located bottom right in the browser (Figure 4).

Like many local office applications, the presentation module provides a number of templates, including various transitions, designs, and formatting options that allow you to create professional presentations. In terms of font management, Docs additionally integrates a large number of fonts, with at least a rudimentary preview. However, it lacks Asian fonts and an option for setting up Asian writing systems.

## Compatibility

Google claims that its office applications are compatible with Microsoft Office. If the documents are simple texts, tables, or presentations, documents stored in Microsoft formats in Google's office applications will look largely identical to how they appear. However, for documents that embed macros in tables or use very complex design elements, there are deviations and sometimes incorrect formatting.

## Disadvantages

Because Google Docs is closely linked to Google services, it also has a number of functional disadvantages: For example, all participants in a session must use a Google account. The Google Docs user invites additional users by email. If more than two users work on a document at the same time, the modification over-

view is lost because Google Docs does not clearly distinguish between the contributions from the individual users. Also, the function of undoing individual modifications is lost if several users collaborate on the same document. For this reason, we recommend limiting the number of active users.

## OnlyOffice

Developed in Latvia by Ascensio System SIA, OnlyOffice [12] (formerly known as TeamLab) is a highly flexible, free office suite under the GNU GPLv3. In addition to the locally installable desktop editors with the usual applications (word processing, spreadsheet, and presentations), OnlyOffice also offers two server-based variants for companies' IT infrastructures.

The free community edition includes three components: a mail and document server, as well as a groupware solution known as Community Server. There is also a commercial Enterprise Edition.

The variety of installation options is striking: The Community Edition is available in the form of precompiled packages in deb and RPM package formats, as well as a Docker image: the Enterprise Edition is also available as a VirtualBox and VMware image, each about 4GB in size [13]. The Community Server lets you manage documents in collaborative environments and transfer files. It

also has a cloud connection to various services and proprietary cloud support. A CRM system including an accounting module, a project management application, an instant messenger, and a calendar with time tracking function are also available.

The Enterprise Edition is also available as 30-day trial version. The Community Server is available for free download on the company's website [14]. The Community Edition is limited to a maximum of 20 users and lacks an integrated backup and restore solution. In addition, the Community Edition lacks automatic updates and a comment function for collaborative document editing.

## System Requirements

The on-premise installation of the OnlyOffice server has some hardware and software requirements. The minimum hardware requirement specified by the manufacturer is a server with a dual-core CPU and 2GHz clock frequency, 2GB RAM, a swapfile of 6GB, and 40GB of available hard disk space. In addition, a Linux kernel version 3.10 or newer must be running on the server.

Depending on the installation method, there are additional software requirements. If you install the server manually, it requires a MySQL database server, a Nginx web server, the Mono framework, and the Node.js runtime environment. If you use Docker images, a Docker environment including daemon is added.

Fortunately, all installation methods are well documented. In addition, OnlyOffice links to YouTube video tutorials via the website.

When installing the server variants, you should make sure that the required packages are installed in as up-to-date versions as possible. Otherwise, the installation may fail.

For example, OnlyOffice under Ubuntu 16.04 LTS does not need the packages provided by Ubuntu, but rather the official Node.js packages from the project side. The versions in the Ubuntu repositories are obsolete, which prevents the OnlyOffice server from being installed. In short, you should strictly follow the instructions to avoid inconsistencies.

## Access

Once the server is installed successfully, you can reach it from every workstation

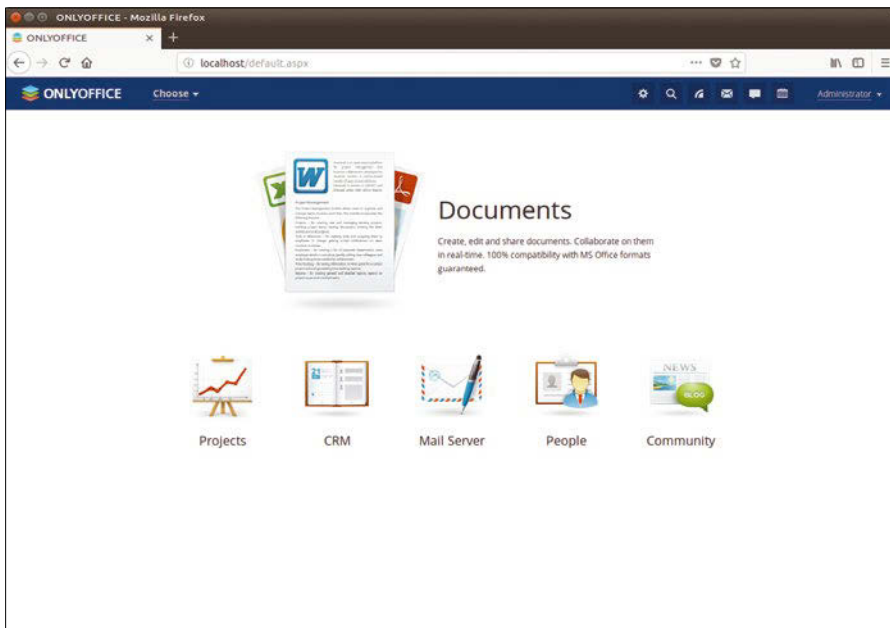


Figure 5: The OnlyOffice interface and its services look pretty tidy.

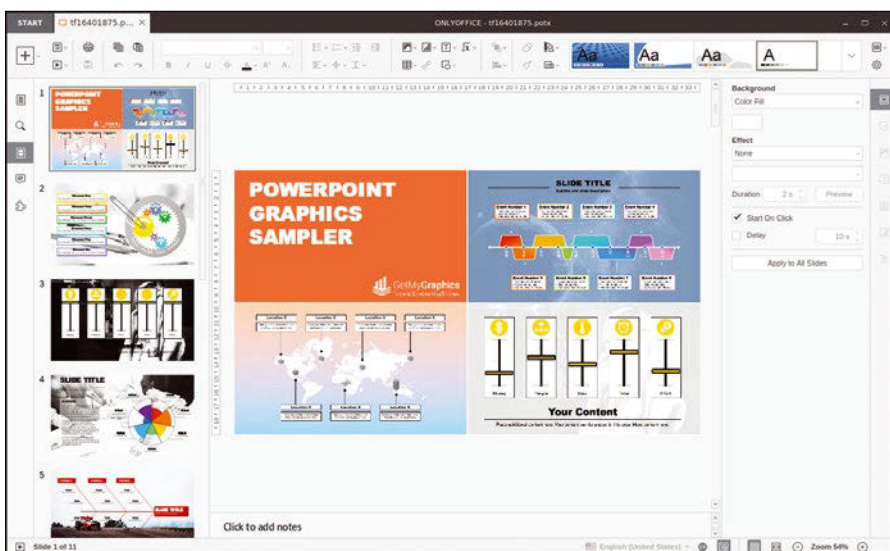


Figure 6: OnlyOffice Desktop Editors using the Community Server.

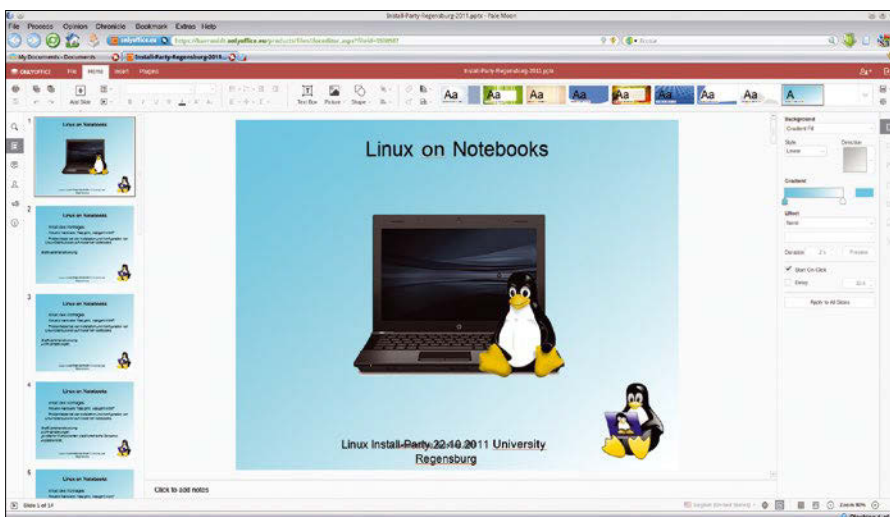


Figure 7: Even simple ODF documents can cause problems for OnlyOffice's conversion filters.

on the intranet. You just need to enter the IP address to access the portal page. Here, you first configure some authentication settings and log on to the server. This takes you to the work screen.

Even a first glance reveals that OnlyOffice, unlike Google Docs or Collabora Office, provides a complete environment for collaborative work. In addition to the document manager, you will find buttons for projects, a CRM system, an email client, symbols for user administration, and community activities on the start page.

All these applications operate under a uniformly designed interface and provide an overview of the basic functions in a superimposed window (Figure 5) when clicked on.

### Office Suite

In on-premise mode, OnlyOffice does not let you edit web-based documents. To edit web-based docs, you'll need to install the document server. A convenient file manager makes it possible to share documents and work on them as a team. However, the actual editing functions of word processing, spreadsheets, or presentation programs are provided by the document server, which needs to be integrated separately. The on-premise mode also offers OnlyOffice Desktop Editors [15] locally on workstations.

The software is licensed under the free AGPLv3 license and, unlike the web-based competitors, is a fully fledged office suite. Various deb and RPM packages are available for Linux, but they only support some Linux derivatives. A portable version (around 217MB in size) that can be used across all distributions is also available. Note that OnlyOffice Desktop Editors is only available for 64-bit hardware.

After downloading, install it locally on the workstations and integrate it into the server services. Once the desktop suite is installed, you can easily import the documents stored on the Community Server into the locally installed office suite and edit them. The advantage of this method is that even documents with very sophisticated formatting can be fully edited.

Important: Other suites, such as LibreOffice, cannot be easily linked to the Community Server. It works, but in heterogeneous environments with Microsoft

formats, this sometimes involves a considerable amount of post-processing.

Due to the sometimes poor conversion filters, LibreOffice returns layout errors for current Microsoft formats even within the Community Server's simple sample documents. OnlyOffice's filters are much more mature in comparison and do not display these errors (Figure 6).

Another drawback for collaborative environments: It is not possible to work on a document simultaneously. However, the Community Server does offer version history for individual documents. In this way, it is at least possible to track which user has edited which document.

## Online

Ascensio System SIA provides the document server separately as free software for download. The company also offers a purely cloud-based solution. It is free of charge for personal use. You are simply required to register and activate your account by email. After logging in, you end up directly in the document manager.

The user interfaces of the on-premise and online versions are identical. In contrast to the server version for intranets, OnlyOffice Online already provides editors for texts, tables, and presentations. On request, you can also integrate cloud services, such as Google, Dropbox, Box, or Yandex.Disk, into OnlyOffice. To integrate a cloud service, click on a start button with a major service provider and authenticate. Locally hosted cloud solutions, such as Nextcloud or ownCloud, implement the connection via the Web-DAV protocol.

In heterogeneous environments, OnlyOffice Online also supports common formats. These formats include the Open Document Foundation file formats used in LibreOffice and OpenOffice, but also PDF, CSV, EPUB, or HTML formats. However, in some cases, the formats come with considerable restrictions.

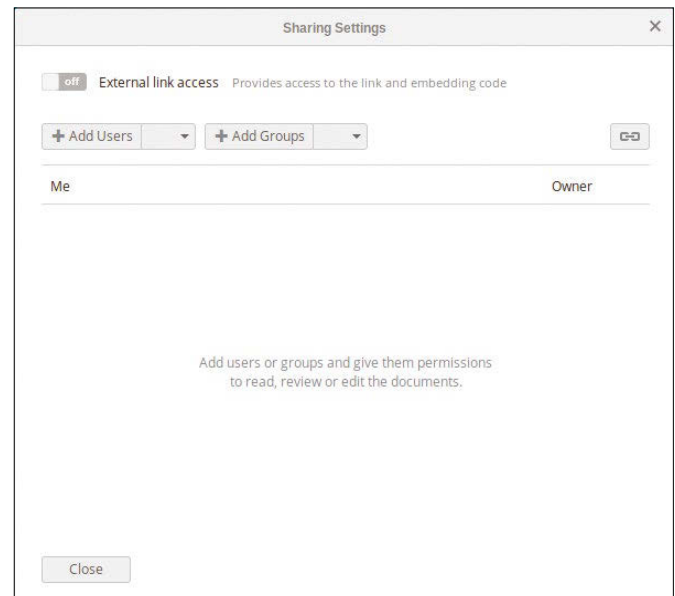
One serious disadvantage of OnlyOffice Online is that the cloud solution forcibly converts documents from a local instance to the current Microsoft Office Open XML format when they are uploaded. It then either stores the file alone or together with the original file in the cloud. However, during the conversion, what if there were already problems in the test with simple docu-

ments in ODF format that required post-processing (Figure 7)?

Even though you may not be allowed to edit all supported formats, the functional scope of the individual editors clearly outshines that of other online solutions. OnlyOffice Online does not let you edit a document simultaneously with two users, but it offers a comment function and an option to share documents. Other users can also work on mobile platforms (Figure 8).

## Pay Barrier

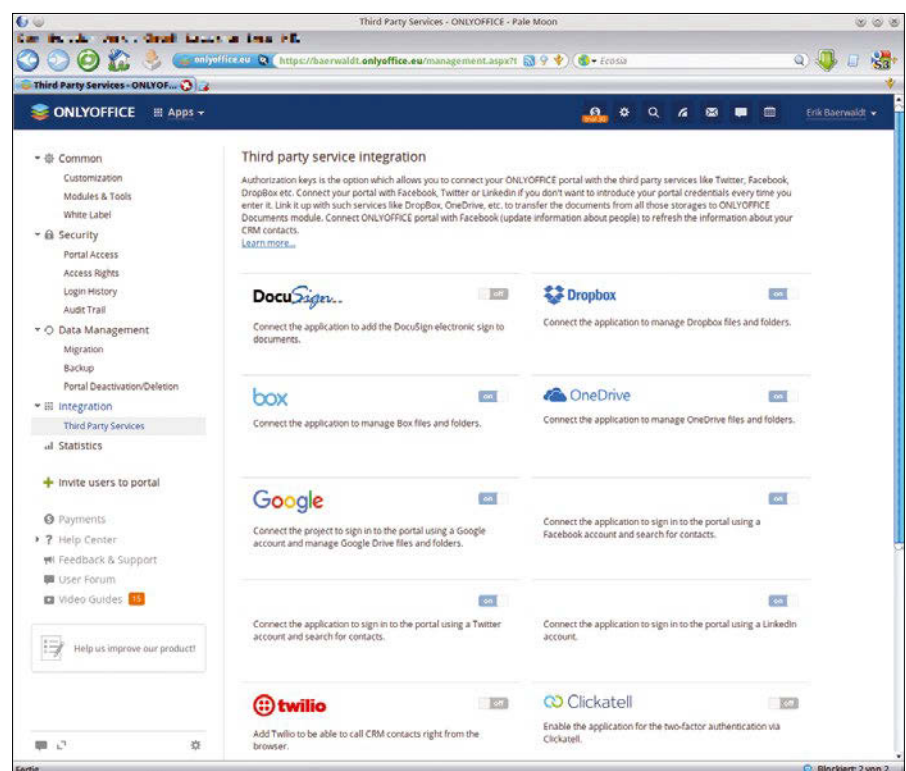
Compared to the dedicated server, the online solution for educational and non-commercial use lacks further modules for collaborative work. Also missing are all additional applications, including the appointment calendar, the CRM system, the email server, and the project manager.



**Figure 8:** In a small dialog box, users release documents for other users and groups.

These services are only provided to commercial users, for whom the company offers a free 30-day trial version. This version has the full functionality of the on-premise enterprise server and is thus able to handle various backup and restore options, as well as the entire spectrum of collaborative work, including chats between individual users.

At the click of a mouse, you can also integrate many third-party services (Figure 9),



**Figure 9:** OnlyOffice can be extended by apps.

including two-factor authentication. If so desired, you can also integrate your contacts from social networks into the CRM module.

### EGroupware

The collaborative solution EGroupware [16], developed in Kaiserslautern, Germany, provides a complete working environment much like OnlyOffice. The EGroupware environment is not limited to pure document processing with the help of an office suite. You can also test the package free of charge for 30 days by creating an account in the cloud. Just enter your email address to activate access.

The provider sets up EGroupware, and the access data is delivered by email. The commercially distributed variants provide various cloud access models to users in Germany. The data are stored in German data centers and are thus also subject to German data protection and security guidelines.

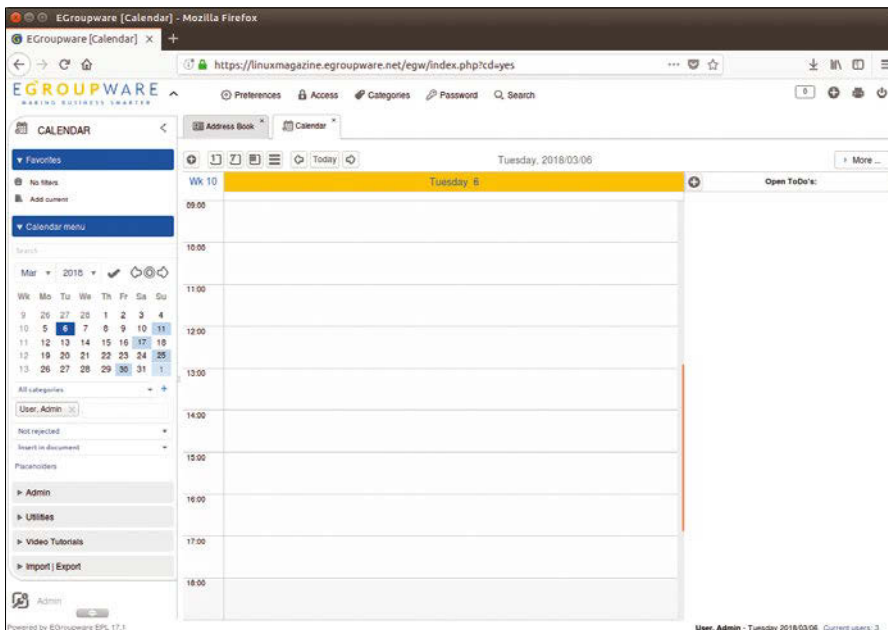
Once you have installed an on-premise solution in either the Pro or Community version [17] or activated the cloud infrastructure, you can access a clearly laid out portal that initially shows you a calendar after successfully logging in. The left-hand sidebar contains the configuration and service options required for first-time users. As the administrator, you can set up new groups and users with the appropriate rights (Figure 10).

Unlike OnlyOffice, EGroupware's office suite does not offer meaningful icons on the portal page. The file manager in the left-hand sidebar in the lower area lets you access collaborative document processing. It displays the existing datasets in list form.

You create documents using the horizontal buttons at the top. Clicking on the plus symbol opens a selection of the document types supported by the integrated Collabora server: In addition to text documents, these types include tables and presentations.

Once you have selected a document type, file name, and file format, the Collabora Office component formats the files in the free ODF format. Alternatively, you can also choose proprietary Microsoft formats, which are limited to the two current variants. The software does not let you save a document as a TXT or PDF file.

After entering the basic file information, the Collabora editor opens in a pop-



**Figure 10:** At first glance, you might notice that the portal also supports collaborative work on documents.

up window. (You must either allow pop-up windows in the web browser or grant an exception in the associated settings dialog.) Alternatively, right-click on the file you created in the file manager and choose *Open*. The Collabora editor then appears in a tab of the web browser.

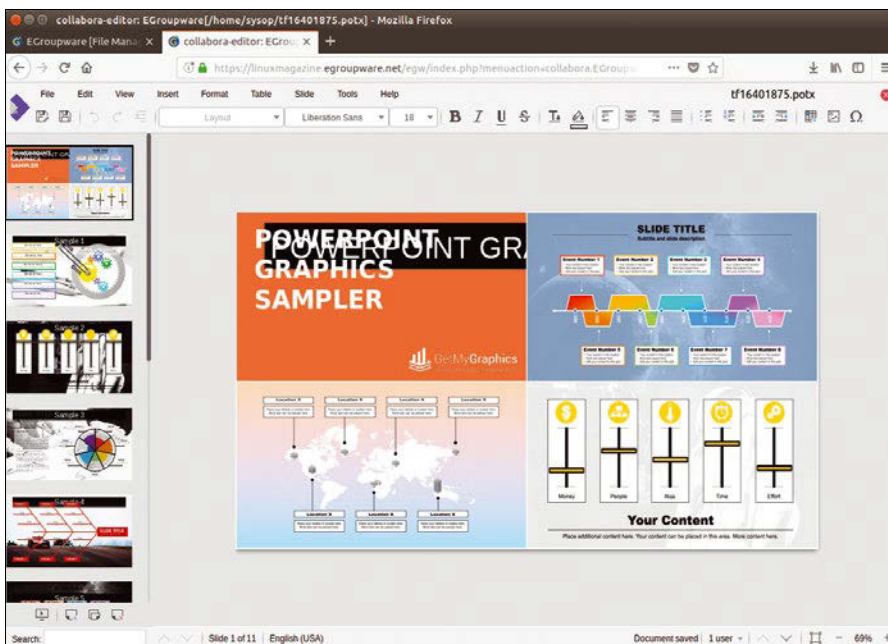
The benefits: You can open multiple files simultaneously, allowing fast access to the file manager or other EGroupware components without navigating through the web browser (Figure 11).

The existing documents also store the cloud variants in the cloud. If re-

quired, you can download documents to the local mass storage device. Use the *File | Download as* dialog, which also allows conversion to PDF format. For other users, the owner of a file sets up shares for collaborative editing by right-clicking the option from the context menu.

### On Premise

Just like OnlyOffice, EGroupware offers a server with the same name for local installation. For local installs, you will find a community version available under the



**Figure 11:** EGroupware opens different tabs for each application.

GNU GPLv2, which is available via the manufacturer's website for open-SUSE, SLES, Fedora, RHEL, Centos, Debian, and Ubuntu [17]. The server is also available in the repositories of some distributions, such as ALT Linux, OpenMandriva, Mageia, and ROSA Linux.

The installation automatically sets up the Apache web server and PHP as dependent packages. In addition, you must manually maintain various add-on modules that you do not initially install into the suite.

The hardware requirements for operating an on-premise server are 8GB RAM, at least 50GB free mass storage, and a 64-bit operating system. Firefox, Chrome, or Safari are the recommended web browsers on clients.

For small businesses, a single server seems sufficient for all components, but the developers advise medium and large organizations to distribute the individual services across multiple servers. Email servers such as Dovecot, Postfix, or Cyrus, as well as the required web servers Apache or Nginx and database servers (we recommend MariaDB or MySQL) can be imported separately. The locally installed community server – as well as the paid enterprise version – offers all groupware functions, but does not have a collaboration server by default. Collaborative document processing is only possible if you integrate a cloud-based or locally installed Collabora server.

## Documentation and Support

The EGroupware suite, which has been around for several years, is well documented. The manufacturers offer tutorials and video tutorials on the Internet, but also hold regular webinars. Enterprise customers can use their own support platform in differently sized packages. You have to create your own username and password. In addition, the chargeable offers include personal support based on fixed time quotas.

## Conclusions

In practice, the two groupware offerings, EGroupware and OnlyOffice, were particularly successful. Both succeed in adding groupware functions to the office suites almost seamlessly. OnlyOffice is ahead by a narrow margin: The suite looks like an ergonomic all-

**Table 1: Overview of Collaborative Office Suites**

	Collabora Online/Office	Google Docs	OnlyOffice	EGroupware
Online server and apps	Yes	Yes	Yes	Yes
On-premise server	Yes	No	Yes	Yes
Office suite used locally	Yes	No	Yes	No
Groupware functions	No	No	Yes	Yes
Import and export function	Yes	Yes	Yes	Yes
ODF support	Yes	Yes	Yes	Yes
Community version	Yes	No	Yes	Yes
Enterprise version	Yes	Yes (integrated in G suite)	Yes	Yes
Integration of foreign cloud services	Yes	No	Yes	No
Fee-based support	Yes	No	Yes	Yes

in-one solution. In addition, the applications are more completely localized in other languages, than those of the duo EGroupware and Collabora.

EGroupware, on the other hand, offers much better documentation in other languages and – together with Collabora – the best support in the test field. In the case of cloud-based services, EGroupware also stands out because the servers run completely in German data centers and users benefit from strict German data protection guidelines.

The purely cloud-based service from Google is also available for slower online access, but only provides basic functions. In addition, it is firmly integrated into the Google universe with non-transparent data protection according to US standards.

One of the drawbacks of all participants is the rather moderate quality of the conversion filters: In EGroupware, Collabora Online sometimes formats incorrectly even simple ODF documents created with OpenOffice. OnlyOffice also sometimes fails on simple ODF documents.

All of the reviewed applications are now functionally mature and fit for productive use (see Table 1). If you want to work cross-platform as part of a team, you will often find that they offer real added value compared to local single-user systems. Nevertheless, a practice-oriented evaluation is recommended before introducing a solution. Otherwise, there may be an additional workload if you have to adapt infrastructure and documents. ■■■

## Info

- [1] MAPI: <https://en.wikipedia.org/wiki/MAPI>
- [2] SaaS: [https://en.wikipedia.org/wiki/Software\\_as\\_a\\_service](https://en.wikipedia.org/wiki/Software_as_a_service)
- [3] EtherPad Lite: <https://yourpart.eu>
- [4] EtherCalc: <https://ethercalc.net>
- [5] Celtx: <https://www.celtx.com>
- [6] Prezi: <https://prezi.com>
- [7] Sozi: <http://sozi.baierouge.fr>
- [8] Collabora: <https://collaboraoffice.com>
- [9] Collabora version comparison: [https://www.collaboraoffice.com/code/#what\\_is\\_code](https://www.collaboraoffice.com/code/#what_is_code)
- [10] Collabora CloudSuite: <https://www.collaboraoffice.com/solutions/collabora-cloudsuite/>
- [11] Google Docs: <https://www.google.com/docs/about/>

- [12] OnlyOffice: <https://www.onlyoffice.com>
- [13] OnlyOffice Enterprise test version: <https://www.onlyoffice.com/enterprise-edition-free.aspx>
- [14] OnlyOffice Community Edition: <https://www.onlyoffice.com/download.aspx>
- [15] OnlyOffice Desktop Editors: <https://www.onlyoffice.com/download-desktop.aspx>
- [16] EGroupware: <https://egroupware.org>
- [17] EGroupware Community packages: <https://software.opensuse.org/download.html?project=server%3AeGroupWare&package=egroupware-ep1>

Exploring the Electron application framework

# Universal Desktop



GitHub's Electron project brings the benefits of web programming to the realm of desktop applications. *By Andreas Möller*

If the boss orders a cross-platform desktop app for internal use, stressed IT staff often lack the time to get familiar with C++/Qt, Java, or Python and the associated application packaging. The Electron project [1] solves this problem by making a desktop app more like a web app. Electron combines Chromium [2] and Node.js [3] into a single run time, thus allowing the developer to build the application using the standard tools of web development, such as HTML, CSS, and JavaScript. The cross-platform nature of web development makes it quite easy to adapt your application to run on Windows, Mac OS, or Linux, and Electron even comes with tools that will package the app for the target OS.

Electron applications even update themselves automatically. Updates are retrieved from known sources, such as a GitHub repository, via an update server.

Electron began in 2013 as the framework for building GitHub's Atom text editor. GitHub open sourced Atom and Electron in 2014. Electron has an MIT license. Version 1.0.0 was released in 2016, and as of this year, it is available in Windows and OS X app stores as well as Linux repositories.

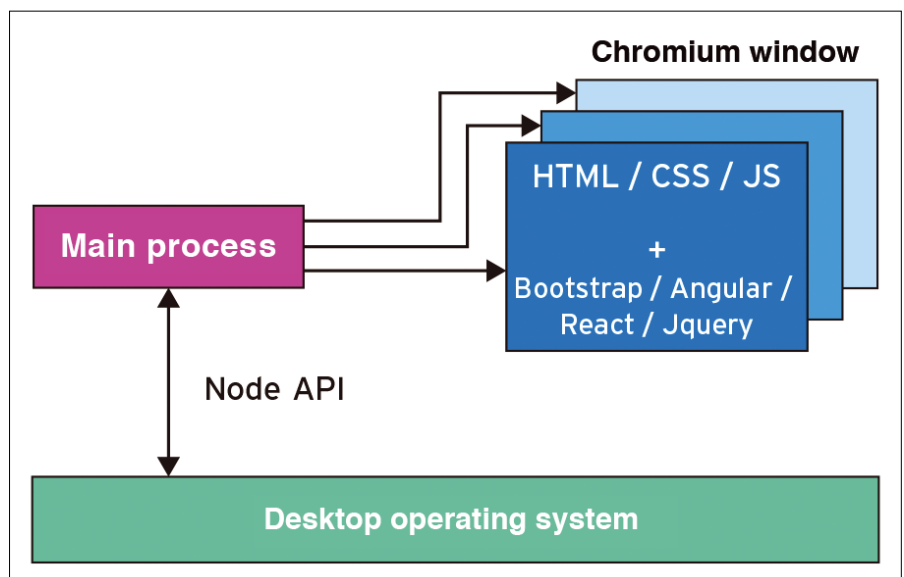
## Bold Design

An Electron app works like a web application but in the reduced browser environment outlined in Figure 1. The Chromium components execute the application (top right in the picture). JavaScript libraries such as Bootstrap [4], React [5], Angular [6], or JQuery [7] contribute to the complete package.

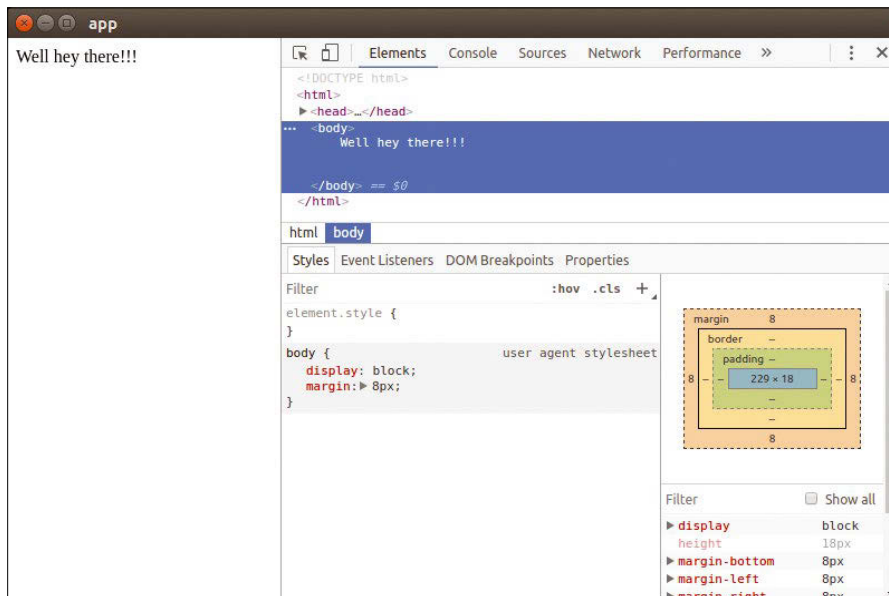
Because the code within the application always executes using the Chromium

run time, Electron avoids the possibility of problems arising from dissimilar browsers or browser versions. Developers also benefit from the debugging tools that come along with Chromium (Figure 2).

The main process manages the browser window. The Node API facilitates the connection to the desktop operating system. The developer can extend the Node environment by adding additional node packages.



**Figure 1:** The main process replaces the browser kernel as a freely programmable framework with a system connection.



**Figure 2:** Electron developers can call upon the help of Chromium debugging tools.

## Installing a Framework

Electron, the boilerplate project, and a set of tools used with Electron, including `electron-packer`, `electron-builder`, or `electron-forge`, all take the form of node packages. Listing 1 installs Electron and all the tools a developer needs to pack the Electron apps onto a Ubuntu 16.04 node. Line 1 installs `git` and `curl` via the package manager; line 3 installs the latest version of Node.js. The necessary package list is created by a Bash script, which `curl` retrieves in line 2, under `/etc/apt/sources.list.d/nodesource.list`, and then updates the list of available packages via `apt-get`.

After `nodejs` in line 3, the script inserts the alternative node package manager, Yarn [8], one line later. With root privileges, Yarn then installs `electron-forge` [9] (line 6).

## Empty Stage

Electron has a number of boilerplate projects [10] that accelerate the development process by providing a minimal executable app out of the box. These

projects internally use `electron-packer` to create the Electron application and `electron-builder` to package platform-specific formats. `electron-forge`, which is installed in Listing 1, offers the best combination of both tools. It also comes from the workshop of the creators of `electron-packer` and `electron-builder`. The `forge` shell command makes use of `electron-forge`.

The life cycle of an app starts when the developer taps the `forge init app` command into the keyboard. This command creates the app project directory from Listing 2 in the style of a boilerplate project. As with any node app, the `node_modules` directory stores all packages listed in the `package.json` configuration file – in the latest version.

Finally, the `src` directory stores the minimal web app, which the developer then converts step-by-step into the desired application.

Initially, the directory contains the `index.html` HTML document and thus the entire web app, as well as the `index.js` file. The programmer checks

the development progress by switching to the app project directory and entering the `forge init start` command. Since a watch process like Angular is missing, a `location.reload()` reloads the app via the debugging console.

Figure 2 shows the started app at run time. The debugging tool appears in the Chromium window on the right side of the screen by default. In the final version, the developer has to remember to disable the debugging tools in the main process code.

The `yarn.lock` file from line 5 in Listing 2 remembers the currently stored packets under `node_modules`. This mechanism makes it possible to reproduce node projects on other machines on a scale of 1 to 1.

## Porting

You can port the boilerplate app from Figure 2 to other desktop systems using the `forge package` command. Without specifying options, the command (under Ubuntu 16.04) creates a standalone version for Linux and the existing x64 processor architecture in the `out/app-linux-x64` subdirectory of the app project folder. In addition to the approximately 80MB executable app in ELF format, this directory also contains other binary files. You will find the web application and the required node packages under `resources/app`. After switching to the directory, the `./app` command starts the app in the shell, which looks exactly like Figure 2.

You can port the app to Windows using the `--platform` command-line switch in the `forge package --platform win32` command. This command creates the app for 64-bit Windows in the `out/app-win32-x64` folder. In addition to the executable `app.exe`, the distribution contains a large number of Windows DLLs and the `resources` folder, which is more or less bit-identical to the matching folder for Linux.

If you copy the folder from Linux to a Windows 10 system and start the

### Listing 1: Installing a Node

```
01 sudo apt-get install git curl
02 sudo curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash -
03 sudo apt-get install -y nodejs
04 npm install yarn
05 sudo su
06 # it also works without Yarn with: sudo npm -g install electron-forge
07 yarn add global electron-forge
```

### Listing 2: Boilerplate App

```
01 |- app
02 |- node_modules
03 |- package.json
04 |- src
05 |- yarn.lock
```

Table 1: Electron Compatibility

System	Version	IA-32	x64	ARMv7l	ARM64
Ubuntu	>=12.04	Yes	Yes (tested)	Yes	Yes
Debian	8	Yes	Yes	Yes	Yes
Fedora	21	Yes	Yes	Yes	Yes
Windows	>=7	Yes	Yes (tested)	Yes	Yes
OS X	>=10.9	No	Yes	No	Yes

Table 2: Possible Package Formats under forge

Type	System	Description
zip	All	Zip archive
squirrel	Windows	Installer for Squirrel-Dot-Windows
appx	Windows	Windows-Store packet
dmg	OS X Darwin	DMG packet
deb	Linux	Debian packet
rpm	Linux	RPM packet
flatpack	Linux	Flatpack packet

app, you won't see any notable differences with Figure 2. Conversely, you could install `electron-forge` under Windows 10, and, in that case, the Linux app created under Windows will run under Linux.

You can assign a processor architecture with the `arch` command-line switch. Table 1 shows the possible combinations of values for `platform` and `arch`.

## Packing

`forge` offers to create platform-specific package formats or installers to facilitate easy distribution of the app (Table 2). `forge make` generates a Debian package for the Boilerplate app. However, the script terminated in our lab while trying to create an RPM package, with the error message from Figure 3. `sudo apt-get install rpm` fixed the problem.

Similar scenes occurred while trying to

create a Windows installer with the `forge make --platform win32` command. In this case, .NET support was missing:

```
sudo apt-get install mono wine
```

Under `out/make/squirrel.Windows/x64`, you will find the `app-1.0.0 Setup.exe` installer file, which we were able to install on Windows 10 without any problems.

Listing 3 displays the configuration object [11] from the `package.json` file as a portal. Lines 2 to 13 contain the package formats to be generated in the `make_targets` sub-object, listed by operating system.

## Conclusions

Electron makes it easy to create desktop apps using popular web technologies such as HTML, CSS, and JavaScript. At

the same time, Electron offers an infrastructure that can port apps to other systems via the command line.

Compared to Cordova Phone Gap [12], the installation is trivial, even if support for mobile operating systems such as Android or iOS is currently missing. If

Listing 3: Configuration of forge

```
01 [...]
02 "forge": {
03   "make_targets": {
04     "win32": [
05       "squirrel"
06     ],
07     "darwin": [
08       "zip"
09     ],
10     "linux": [
11       "deb",
12       "rpm"
13     ]
14   },
15   "electronPackagerConfig": {
16     "packageManager": "yarn"
17   },
18   "electronWinstallerConfig": {
19     "name": "app"
20   },
21   "electronInstallerDebian": {},
22   "electronInstallerRedhat": {},
23   "github_repository": {
24     "owner": "",
25     "name": ""
26   },
27   "windowsStoreConfig": {
28     "packageName": "",
29     "name": "app"
30   }
31 }
32 [...]
```

you want to be on the safe side, simply pack on the target system each time and test the app immediately. ■■■

## Info

- [1] Electron: <https://electronjs.org>
- [2] Chromium: <https://www.chromium.org>
- [3] Node.js: <https://nodejs.org>
- [4] Bootstrap: <https://getbootstrap.com>
- [5] React: <https://reactjs.org>
- [6] Angular: <https://angular.io>
- [7] JQuery: <https://jquery.com>
- [8] Yarn: <https://yarnpkg.com>
- [9] electron-forge: <https://github.com/electron-userland/electron-forge>
- [10] Boilerplate projects: <https://github.com/sindresorhus/awesome-electron#boilerplates>
- [11] Configuration of electron-forge: <https://github.com/electron-userland/electron-forge#config>
- [12] Phone Gap: <https://phonegap.com>

Figure 3: Starting `rpmbuild` as a separate process using `spawn` failed at first because `rpm` was not yet installed.



# Complete Your Open Source Library with Archive DVDs!

Each fully-searchable archive DVD includes past articles so you can find the content you need quickly.

**Save hundreds off the print and digital copy rate with a convenient archive DVD!**



**Order Your DVD Now!**  
[shop.linuxnewmedia.com](http://shop.linuxnewmedia.com)

Your files will be assimilated

# Borg Backup

In Star Trek, the Borg adds individuals to its collective consciousness, an apt metaphor for any backup application that stores individual files in an archive. BorgBackup creates folder repositories for multiple archives, making it an especially befitting description of assimilation. *By Bruce Byfield*

**B**orgBackup [1] was originally a fork of Attic [2], begun in 2015 by developers who wanted faster-paced development. Attic is no longer in development, although stable versions remain available in Debian and several other distributions. Meanwhile, BorgBackup has lived up to its mandate, not only enhancing the Attic code, but also gaining a reputation for its innovative approach. Although other compression and backup utilities have encryption, incremental, and remote options, what distinguishes BorgBackup is how it implements such features,

## 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>

making them more visible and easier to manipulate. For example, because an incremental backup is in a separate file from the original, it can be retrieved by itself. In many ways, BorgBackup is a modernization of long-time Linux archivers such as tar, gzip, and bzip2, offering increased efficiency and speed with its choice of compression formats and its own set of options. In addition, BorgBackup offers fine-tuned control of features, such as the compression level and the details stored in its logs, while using intelligent defaults that allow users to quickly get up and running.

BorgBackup's documentation [3] does warn that the application requires a reasonable amount of free space to operate. The exact amount, of course, depends on the size of the files it uses, but at least 2GB should be available. However, to ensure trouble-free backups, repositories should be created on filesystems with plenty of free space.

## The Basics

Like an increasing number of modern command-line applications, BorgBackup's command structure consists of three parts: the basic command; a sub-command, such as create, extract, or delete; and the options and paths to a file. Many options can be used with more than one of the sub-commands and can be placed either after the sub-command or after the paths, although to avoid any confusion, users should probably use one position consistently.

Before archiving, BorgBackup requires the creation of a repository in which to store backups. This repository can be protected by a passphrase, as well as encrypted. The arrangement has the advantage of reducing the number of passphrases to remember, even though, technically, individual or additional passphrases would be more secure in some cases.

To create a repository, enter:

```
borg init PATH
```

```
bb@nanday:~/fonts$ borg init ./free-fonts
Enter new passphrase:
Enter same passphrase again:
Do you want your passphrase to be displayed for verification? [yN]: y
Your passphrase (between double-quotes): "abc123"
Make sure the passphrase displayed above is exactly what you wanted.

By default repositories initialized with this version will produce security
errors if written to with an older version (up to and including Borg 1.0.8).

If you want to use these older versions, you can disable the check by running:
borg upgrade --disable-tam 'free-fonts'

See https://borgbackup.readthedocs.io/en/stable/changes.html#pre-1-0-9-manifest-spoofing-vulnerability for details about the security implications.
bb@nanday:~/fonts$ █
```

**Figure 1:** To back up files, you must first create a repository. Viewing the new repository's passphrase can help you remember it or encourage you to reconsider a weak password.

```
bb@nanday:~/fonts$ borg list ./daily::freefonts
Enter passphrase for key /home/bb/fonts/daily:
drwx----- bb      bb           0 Thu, 2016-04-21 18:25:17 Gentium
drwx----- bb      bb           0 Thu, 2016-04-21 18:25:18 Gentium/GentiumPlus-1.510
-rwx----- bb      bb      9108 Thu, 2016-04-21 18:25:18 Gentium/GentiumPlus-1.510/F
ONTLOG.txt
-rwx----- bb      bb      8393 Thu, 2016-04-21 18:25:18 Gentium/GentiumPlus-1.510/G
```

**Figure 2:** The `list` sub-command displays the contents of an archive.

This command automatically creates a repository, prompting for a passphrase (Figure 1). BorgBackup does not prevent a weak passphrase. However, if you choose the option to display the password before continuing, it will give you the chance to reconsider your choice. The path can be relative or absolute, as well as local or remote and connected by SSH (`ssh://PATH:2022/; USER@HOST:PATHcs`). If necessary, a remote or external path can be activated using the `mount` sub-command, most of whose options are contained in the Filesystem in Userspace (FUSE) man pages.

In most cases, too, you may want to include the `--encryption=repokey` option, which stores the passphrase in the repository, or else the `--encryption=keyfile` option, which stores the passphrase in `~/.config/borg/keys`. The `keyfile` option is most secure if your home directory is on a separate device from the repository. Later, the sub-command `change-passphrase` can be used to change the passphrase, although if the repository has been compromised, your data is more secure if you create an entirely new one.

In either case, the passphrase uses 256-bit AES encryption and is verified

with HMAC-SHA256. The passphrase is stored in UTF-8 and should avoid any characters that have a specific meaning at the command line, such as a forward slash (/) or a regular expression like an asterisk (\*).

After a repository is created, you can add files to it with the `create` sub-command:

```
borg create REPO::ARCHIVE SOURCES
```

Multiple sources can be specified in a space-separated list. You can create an incremental backup by specifying sources already archived, which will substantially reduce the amount of time required to update an archive. Once a repository and archives are created, you can use other sub-commands for basic functions (see Table 1).

## Advanced Options for Creating Archives

When using the `create` sub-command, one of the most important settings is compression. By default, BorgBackup uses the LZ4 format, a fast choice, but one with a relatively low compression rate. If you have the disk space, you may prefer to specify `--compression none (-C)`. Better control over compression can be had by specifying another format, each of which has its own scale for the level

**Table 1:** Selected BorgBackup Sub-Commands

Sub-Command	Description
<code>list</code>	Displays archives in a repository or an archive (Figure 2).
<code>delete</code>	Removes an archive.
<code>extract:</code>	Restore archives.
<code>prune</code>	Sets a pattern for regularly removing archives and their contents by specifying by prefix or setting what to keep with the options <code>--keep-daily</code> , <code>--keep-weekly</code> , and <code>--keep-monthly</code> . Works well with automated operation (see the "Automating the Command" section).
<code>rename</code>	Gives an existing archive a new name.
<code>diff</code>	Compares two archives.
<code>key change-passphrase</code>	Changes a repository's passphrase.
<code>key export</code>	Creates a file to store the passkey.
<code>key import</code>	Retrieves an exported passkey.
<code>export-tar</code>	Creates a tar version of the archive.
<code>config</code>	Edits the name and key value for the specified repository.

of compression: `zstd` (1-22), `zlib` (0-9), and `lzma` (0-9). In all of these scales, the lowest number gives the least compression but runs the quickest, and, if a compression level is not specified, BorgBackup uses a medium level. The format and compression level can be set using the option:

```
--compression FORMAT LEVEL
```

The preferable format and compression level is a matter for expert debate, but keep an eye on CPU activity and free disk space while running each format to help decide which one to use in a particular circumstance.

Unless the `--verbose` (`-v`) and `--show-rc` options are used, BorgBackup gives minimal output to avoid clutter. The output it gives is enough in routine circumstances and often includes helpful suggestions. However, at times you may need to look at the logfile for more detailed information. No logfile is written by default, but you can specify one as an environment setting (see “Automating the Command”), which may be advisable on a network or when dealing with any important files. You can set what is logged by creating an archive with one of four options:

- `--debug`: Logs everything in detail.
- `--warning`: Logs warning, error, and critical-level messages.
- `--error`: Logs error and critical-level messages.
- `--critical`: Logs only critical-level messages

While it might seem wisest to set the log information to `--debug`, too much information might make the log more difficult to read, so you should experiment to discover what level of information you actually need.

Other options are also available. The option `--exclude PATH` can be used to exclude items from an archive or else the backup can be limited to one filesystem with:

```
--one-file-system (-x)
```

```
bb@nanday:~/fonts$ borg check ./free-fonts::Lato
Enter passphrase for key /home/bb/fonts/free-fonts:
bb@nanday:~/fonts$ borg check --verbose ./free-fonts::Lato
Starting repository check
Completed repository check, no problems found.
Starting archive consistency check...
Enter passphrase for key /home/bb/fonts/free-fonts:
Orphaned objects check skipped (needs all archives checked).
Archive consistency check complete, no problems found.
```

Figure 3: The check sub-command tests that an archive is usable.

the default `0077` permissions can be changed with:

```
--umask OCTAL-PERMISSIONS
```

The `atime` attribute, which records the last time a file was accessed, can be omitted with `--noatime`; the `ctime` attribute, which records the last time the file was changed, with `--noctime`; and the user and group identifiers can be restricted to numeric identifiers with `--numeric-owner`.

All these options can make the initially simple `borg` command considerably more complex, so, having set up a command, you might want to run:

```
--dry-run (-n)
```

to test your carefully crafted command before actually running it. Upon creating an archive, you should also run the `check` sub-command (Figure 3) to ensure that the archive is reliable. After all, the very worst time to find that an archive is unusable is when you need to extract it.

## Automating the Command

Another way to make using a complex command easier is to add the most commonly used options as environment variables using the `export` command [4]. The `borg` command will use the variables defined as part of the environment unless alternatives are specified. For example, `BORG_REPO=PATH` sets the location of the default repository, while `BORG_PASSPHRASE=PASSPHRASE` gives the passphrase for the default repository. Other possible environment variables store responses to warnings, such as

```
BORG_UNKNOWN_UNENCRYPTED_REPO_ACCESS_IS_OK=yes
```

which responds to the warning, “Attempting to access a previously un-

known unencrypted repository.” As you might expect, none of the available environment variables will function if you use the `sudo` command to operate as another user, since the change of user is generally accompanied by a change in environment.

You can also use Bash scripts to automate backups and then use a crontab to run the script. BorgBackup’s Quick Start provides an example of such a script that can be modified to suit your system [5].

## Going on from Here

As you can tell by the number of sub-commands, there is much more to BorgBackup than I have presented here. I have given an overview of basic considerations, plus enough additional material to use BorgBackup with some efficiency. If you want to learn more, I suggest not using the rather verbose man page, but going instead to the online documentation, which is divided into a Quick Start and more advanced instructions.

BorgBackup is just different enough from other archive tools that it takes a while to learn. However, once learned, its structure is quick and efficient. Take the time to understand it, and you might just decide that BorgBackup is your backup utility of choice – even if you are accustomed to a GUI. ■■■

## Info

- [1] BorgBackup: <https://www.borgbackup.org>
- [2] Attic: [https://en.wikipedia.org/wiki/Attic\\_\(backup\\_software\)#Borg](https://en.wikipedia.org/wiki/Attic_(backup_software)#Borg)
- [3] BorgBackup documentation: <https://borgbackup.readthedocs.io/en/stable>
- [4] BorgBackup environment variables: <https://www.mankier.com/1/borg>
- [5] Backup script: <https://borgbackup.readthedocs.io/en/stable/quickstart.html#automating-backups>

# REAL SOLUTIONS for REAL NETWORKS

ADMIN – your source for technical solutions to real-world problems.

Learn the latest techniques for better:

- network security
- system management
- troubleshooting
- performance tuning
- virtualization
- cloud computing

on Windows, Linux, and popular varieties of Unix.

**GET IT FAST**  
with a digital subscription!

**6 issues per year!**

**ORDER NOW**

[shop.linuxnewmedia.com](http://shop.linuxnewmedia.com)



Determining assets across a shrinking number of financial APIs

# Bargain Hunter

Financial wizard Mike Schilli is annoyed that some web services have discontinued serving up real-time stock market data. To keep an eye on his investment dollars, Mike taps into a little known interface for stock prices. *By Mike Schilli*

**F**or users of stock-quote-serving APIs, the fruit is currently hanging higher. First, Google discontinued its web service [1] for

real-time stock market data (Figure 1); then, Yahoo followed suit and pulled the plug. With the end of their undocumented but widely used share price interface in CSV format [2], Yahoo, perhaps unknowingly, also pulled the rug from under a number of open source projects, such as the Perl `Yahoo::Finance` CPAN module and the Python `yahoo-finance` package. A few open source portfolio-tracking applications using these popular packages were dragged into the abyss along with them.

On the GitHub sites of these projects, the *Issues* tabs are now overflowing with comments, there's quite a bit of wailing and gnashing of teeth from users, and much wringing of

hands of those who hope to find a replacement for the vanished free real-time (or at least time-delayed) stock market price data source.

## Free Is Cheap Enough

For a hefty monthly fee, of course, you can find dozens of business-level stock market data providers, such as Bloomberg, but only a few competitors offer free quotes, often only to registered users who are adding an issued API key to every request. It is probably only a matter of time before the free source presented today also dries up, but I'm working on the assumption that, in the future, options for retrieving the data free of charge will still be around.

Developers should therefore be prepared for the fact that an application's data back end can change at short notice. Thanks to a design abstracting away the share price source, this should be easy to handle with a simple upgrade of the quote library for applications; the caravan of thirsty camels, after only find-

## 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](mailto:mschilli@perlmeister.com) he will gladly answer any questions.



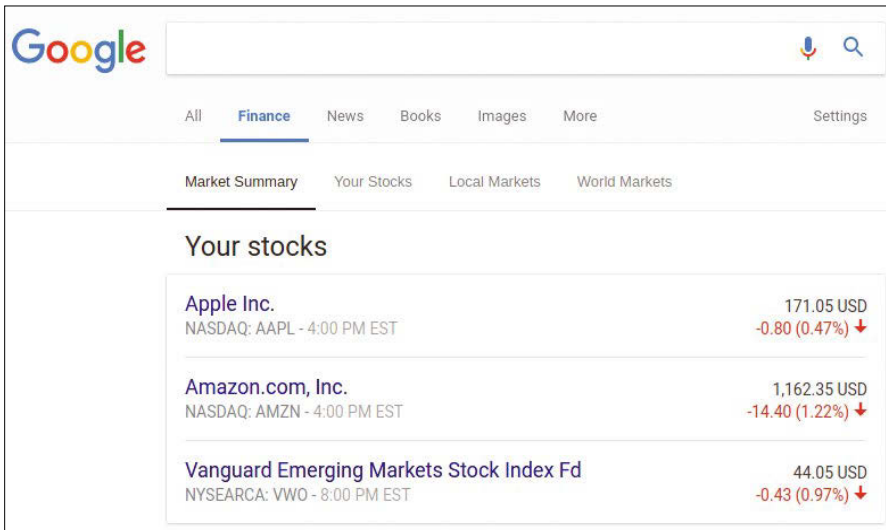


Figure 1: Google's financial site for portfolio tracking has seen better days.

ing dried out water holes at the old oasis, will move on to more refreshing destinations.

Listing 1 [3] uses a still open Yahoo source to obtain the current market prices of equities on the world's major stock exchanges. For example, the *amzn* ticker symbol in dollars reveals the daily value of the Amazon share on the New York Stock Exchange. Those who want to explore Amazon's share price on the German Xetra Stock Exchange in euros enter *amz.de* instead.

To do this, the `quote()` function awaits the ticker symbol and sends a request to the web API that returns a data structure encoded in JSON. Under the `regularMarketPrice` key is the numerical value for the price in the currency customary on the market where it is traded.

## Cash Check in the Money Bin

Banks and stock brokerage houses offer portfolio management functions and calculate the net value of the deposit for their clients. Google and Yahoo also offer portfolios, although they are in the process of change [2]; both seem to be looking for new ways to make money off the service somehow.

The advantage of a home-grown script is that it adds all items across accounts or brokerages. For this purpose, the user has to keep a YAML file with the items in their portfolio up to date manually, as shown in Listing 2. However, this approach preserves the investors' privacy more reliably than third-party applications that have a monetary interest in mining user data and using it for marketing purposes or, worse, selling it to the highest bidder.

Listing 2 shows two fictitious accounts, one at Wells Fargo and one at Citibank, with 100 Amazon and 50 Google shares, the latter of which are now traded under the "Alphabet" name, although the company's ticker symbol on Nasdaq is still *goog*. Both accounts also have a Cash item. The `mymoney.py` script in Listing 3 adds the cash at face value but determines the current value

### Listing 1: myquote.py

```
01 #!/usr/bin/env python
02 import requests
03 import json
04
05 def quote(symbol):
06     r = requests.get(
07         'https://query1.finance.yahoo.com'+
08         '/v7/finance/quote?symbols=%s'
09         % symbol)
10     data = json.loads(r.text)
11     return (data['quoteResponse']
12           ['result'][0]
13           ['regularMarketPrice'])
```

of the share items via the API and multiplies the stock price by the number of shares in the deposit.

To perform this calculation, Listing 4 (line 3) includes the `yaml` module, which you must have installed previously by typing:

```
pip install --user pyyaml
```

It provides the `load()` method used in line 8 to parse the account data from the portfolio file in Listing 2. Line 30 iterates across the accounts found; line 32 iterates across the items stored therein. In the current version, the script supports cash (`cash`) and shares (`stock`) but could easily be extended to include bonds, options, or items in foreign currencies, whose exchange rate the script then can retrieve in a similar fashion at run time. The `cash_dump()` function from line 18 determines the value of a cash item. There's no magic being performed here; it only returns a tuple with the name `Cash` and the amount.

### Listing 2: portfolio.yaml

```
01 - account: Wells Fargo
02   holdings:
03     - type: stock
04       ticker: amzn
05       shares: 100
06     - type: cash
07       amount: 5170.99
08
09 - account: Chase Bank
10   holdings:
11     - type: stock
12       ticker: goog
13       shares: 50
14     - type: cash
15       amount: 1234.56
```

### Listing 3: mymoney.py Calculations

```
$ ./mymoney.py
Wells Fargo
  100 x amzn: 148292.00
    Cash:    5170.99
Chase Bank
  50 x goog:  55567.00
    Cash:    1234.56
Total: 210264.54
```

Listing 4: mymoney.py

```

01 #!/usr/bin/env python
02 import requests
03 import yaml
04 import myquote
05
06 def pf_read(filename):
07     with open(filename, 'r') as stream:
08         return yaml.load(stream)
09
10 def amount_fmt(amount):
11     return "%10.2f" % amount
12
13 def stock_dump(ticker, shares):
14     price=myquote.quote(ticker)
15     return("%d x %s" % (shares, ticker),
16           price*shares)
17
18 def cash_dump(amount):
19     return "Cash", amount
20
21 def pos_dump(pos):
22     if pos['type'] == 'stock':
23         return stock_dump(pos['ticker'],
24                           pos['shares'])
25     elif pos['type'] == 'cash':
26         return cash_dump(pos['amount'])
27
28 total=0
29
30 for account in pf_read("portfolio.yaml"):
31     print(account['account'])
32     for pos in account['holdings']:
33         (name, amount) = pos_dump(pos)
34         total += int(amount*100)
35         print("%15s: %s" %
36               (name, amount_fmt(amount)))
37
38 print("\n%15s: %s" %
39       ("Total", amount_fmt(total/100.0)))

```

To determine the value of a stock item, on the other hand, Listing 4 uses the `myquote.py` module from Listing 1 and calls its `quote()` function with the ticker symbol as a parameter. It multiplies the returned market price by the number of shares in the securities account and feeds this back to the main program together with a table key, all wrapped into a tuple. The main program only needs to add the line items across all accounts and output the total net value for the user (Listing 3).

## Rounding, but Correctly

One thing to keep in mind is that computers are notoriously bad at adding floating-point numbers. If you add 1.23 and 4.56 together, you may get a result that deviates from the expected value of 5.79, because the CPU does not represent 1.23 exactly as 1 and 23 hundredths, but approximates it in binary format. In a past article, I discussed how even eBay can't add properly and that their monthly bill regularly is off by a few cents [4].

When it comes to adding money amounts, it is smarter to compute in cents rather than dollars. In other words, simply multiply the dollar amount by 100 and add it as a rounded integer; when displaying the result, inject a decimal point in front

of the last two digits – Listing 4 does just that.

## Predicting the Future

With the relatively easily available daily price data of known stocks, you could fall into the trap of feeding an artificial intelligence network historical stock price data and trying to predict future price jumps. Then, armed with this knowledge, you could make a massive investment and sit back and wait for payday. Ha! However, the timeless arguments from the bestseller *A Random Walk down Wall Street* [5], which are no longer entirely new but all still valid, speak against this, stating that, in this case, the millions of market participants with similar ideas would quickly close any wormholes in the time structure, if they actually existed, because they would have known about possible price jumps at the same time and would have reaped the profits way before the crusading private investor.

There's no easy money to be made at the exchanges, I'm afraid, or is there? Are the exchanges just glorified casinos and follow the "random walk" theory [6], not because their development is actually random, but because prices are unpredictable? You're free to prove the random walker disciples

wrong, of course. Try it; it could be lucrative, if it works! ■■■

## Info

- [1] "Google Finance as you know it is going away": <https://www.marketwatch.com/story/google-finance-as-you-know-it-is-going-away-here-is-what-will-remain-2017-09-27>
- [2] "Has Yahoo finance web service disappeared?": <https://stackoverflow.com/questions/38355075>
- [3] Listings for this article: <ftp://ftp.linux-magazine.com/pub/listings/linux-magazine.com/210/>
- [4] "Watchful Seller" by Michael Schilli, *Linux Magazine*, issue 188, July 2016, <http://www.linux-magazine.com/Issues/2016/188/Perl-Monitoring-Payments>
- [5] Malkiel, Burton G. *A Random Walk down Wall Street: The Time-Tested Strategy for Successful Investing*. W. W. Norton & Company, 2016: <https://www.amazon.com/Random-Walk-down-Wall-Street/dp/0393352242>
- [6] Random walk theory: [https://en.wikipedia.org/wiki/Random\\_walk](https://en.wikipedia.org/wiki/Random_walk)

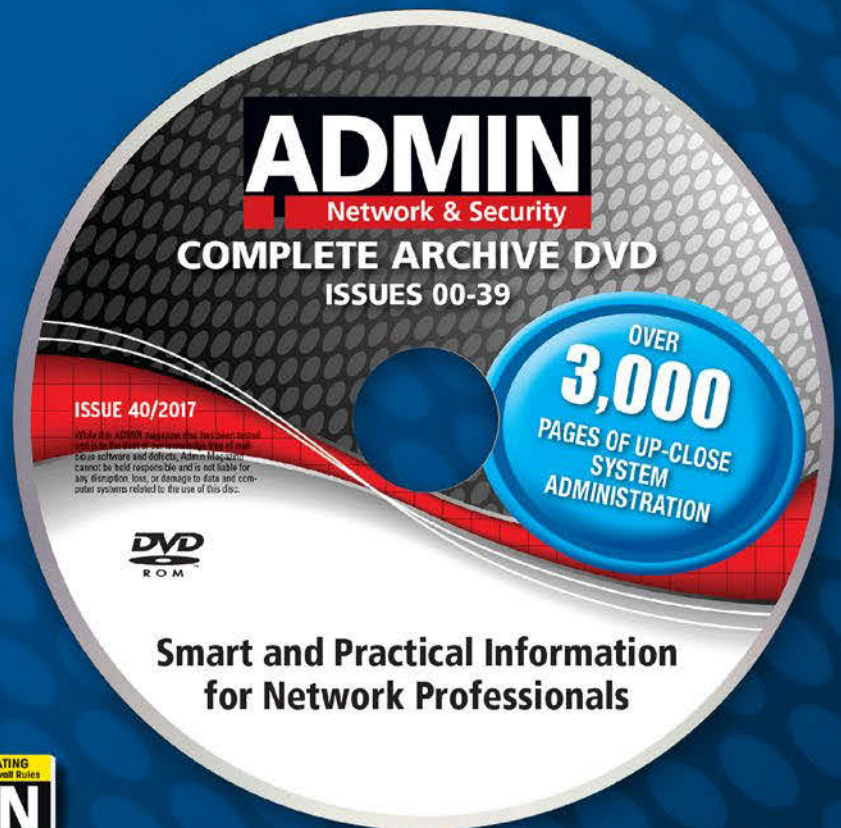


Find us on  
**Facebook**

<http://www.facebook.com/linuxpromagazine>



# 7 Years of ADMIN on One DVD



This searchable DVD gives you 40 issues of ADMIN, the #1 source for:

- systems administration
- security
- monitoring
- databases
- and more!

Clear off your bookshelf and complete your ADMIN library with this powerful DVD!

**ORDER NOW!**  
[shop.linuxnewmedia.com](http://shop.linuxnewmedia.com)





## Blogging with a state-of-the-art WordPress alternative

# FRIENDLY GHOST

Blogging usually implies WordPress, but Ghost presents a serious challenge. *By Christoph Langner*

**A**ccording to an analysis by W3Techs [1], neither Google nor Amazon control the Internet – the WordPress content management system (CMS) does. They say the open source CMS now drives 30 percent of all websites, which translates to a market share of just over 60 percent. Thus, you are unlikely to go a day without stumbling across a WordPress page while browsing.

Even though most web developers are looking for one of the usual candidates from the CMS trio of WordPress, Joomla, and Drupal, you should take a look at the CMS inhabiting the bottom of the W3Techs list. The still quite new Ghost [2] has only secured a small share of the market, but its modern approach and its active community make the CMS interesting for open source-minded bloggers.

### Ghostly

In contrast to WordPress, the Ghost CMS does not rely on PHP, but on JavaScript and Node.js, which makes it a bit difficult to get started. However, the modern web platform plays to its strengths. The idea for Ghost came from Hannah Wolfe and John O’Nolan, who gained great experience in CMS as the former deputy head of design of the WordPress team. O’Nolan no longer sees WordPress as the simple tool for bloggers that it once was, but as a platform for all kinds of web projects [3]. With a handful of plugins, you can set up virtually any project on the web, from a bug tracker to a web shop to an online community.

Its great flexibility is also the biggest weakness of WordPress: The CMS has lost its focus on the original target group, bloggers, which are exactly who Ghost is

now targeting. The project was given a significant boost by a successful Kickstarter campaign. In just under a month, around £200,000 came together, which was far more than the £25,000 funding target [4].

The Ghost Foundation emerged from the accumulated capital and developed and managed the code under the MIT license as a non-profit organization. The foundation currently employs eight people, and the steadily growing community contributes time and code to the project. As with WordPress, the Ghost CMS is available for self-hosting, or Ghost(Pro) [5] is offered as a hosted version for a fee starting at \$19 a month.

### Drafted

Ghost recommends Ubuntu 16.04 with 1GB of RAM, MySQL as the database back end, and Nginx as the web server.

Lead image, toa-heftiba, Unsplash.com

**Listing 1: Activating Web Server**

```
$ sudo apt update
$ sudo apt full-upgrade
$ sudo apt install nginx mysql-server
$ sudo ufw allow 'Nginx Full'
```

As a JavaScript application based on Node.js [6], Ghost provides its own web server. Node.js listens for requests on port 1024 or greater so that the service does not have to run with root privileges, and Nginx accepts requests on ports 80 and 443 (for SSL) as a reverse proxy before passing them on to Ghost. In a test on Ubuntu Server 17.10, Ghost proved in our lab that it can handle a state-of-the-art software stack.

As a prerequisite, we assume that the server can be reached over an Internet domain. In the case of a home server connected to the Internet via DSL or cable, you must ensure that the URL is accessible via DynDNS and forward ports 80 and 443 from the router to the server. Refer to the Installation Guide [7] to update the server and install Nginx and MySQL. The system requests a root password for the SQL server (which you must remember); then, you need to activate the web server in the firewall (Listing 1).

**Nodes**

Ghost developers recommend using the 6.x LTS branch of Node.js. Ubuntu 16.04 and newer is the easiest way to load it via *nodesource.com*. The installation script inserts a package source into the system, from which it then installs the application. With the node package manager (npm), you can install the ghost-cli setup wizard:

```
$ wget https://deb.nodesource.com/setup_6.x
$ chmod +x setup_6.x
$ sudo ./setup_6.x
$ sudo apt install nodejs
$ sudo npm i -g ghost-cli
```

At first glance, this procedure might seem complicated, but it saves much time and effort in the long run.

At this point, you have created the necessary prerequisites to put the blog system on the server:

```
$ sudo mkdir -p /var/www/ghost
$ sudo chown <User>:<Group> /var/www/ghost
$ cd /var/www/ghost
$ LANG=C ghost install
```

The first line creates the `/var/www/ghost/` working directory; the second

line assigns the user rights to the folder. Next, you switch to the newly created directory, which must be empty, and run the installation wizard. The `LANG=C` changes the language settings to the default locale; otherwise, the script runs into an error and terminates.

The installation routine first wants to know the blog URL, the address of the MySQL server, the root login, and the previously assigned MySQL password, as well as a name for the database. You can enter the URL to access the blog later. If the blog and database are on the same computer, the address for the MySQL server is *localhost*. The database does not have to exist: The setup creates it, if necessary.

The Ghost command-line interface (CLI) handles most of the maintenance work and the installation. You can install the current version with `ghost update`. Alternatively, take a look at what's going on behind the scenes with `ghost log`. For information on all options of the Ghost client for the command line, enter `ghost help`.

**Go to Blogging**

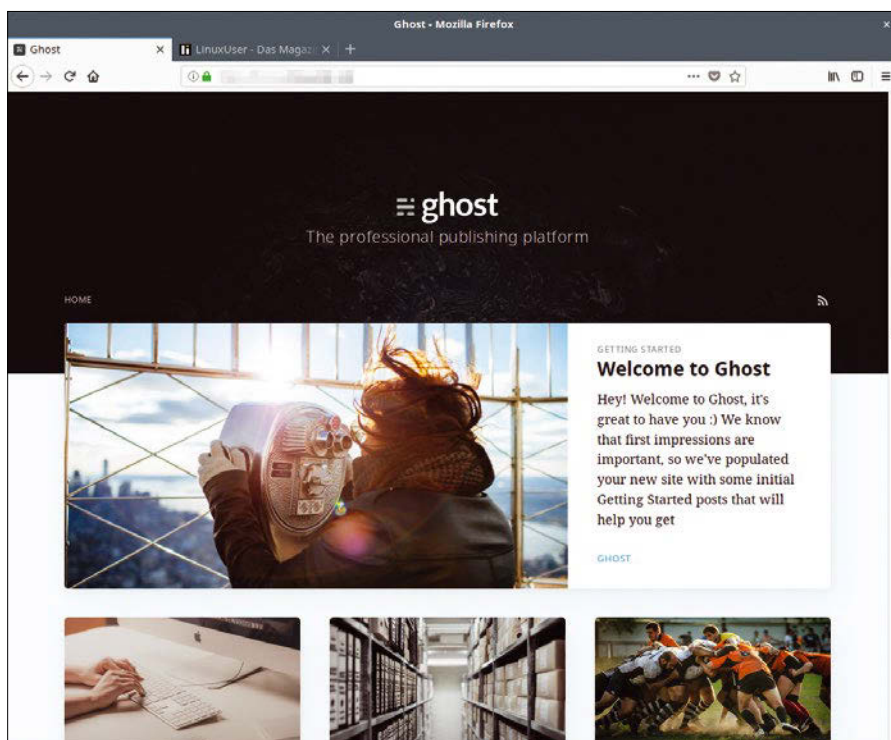
The script takes over configuration of the Nginx web server. If necessary, you can encrypt the connection using a Let's Encrypt SSL certificate. In this case, in addition to port 80, port 443 must be forwarded to the home server. Finally, let systemd control Ghost and then start the service. If everything works fine, you can reach the blog in your browser under `http://<URL>`; if SSL is active, the system automatically redirects to `https://<URL>` (Figure 1).

You can load the front end for administration in `http://<URL>/ghost`. The first time you visit the site, create a user and give the blog a name; then, open *Settings | General* and set the local time under *Timezone*.

The preinstalled content demonstrates the features of the blog. If necessary, delete all content with *Labs | Delete all Content*. The system sends messages (e.g., when resetting passwords) via Nodemailer [8]; you can check the function with *Labs | Send a test email*.

**Functions**

When composing an article, Ghost deliberately avoids using a WYSIWYG editor



**Figure 1:** The Ghost CMS system is based on JavaScript and Node.js and optimized for bloggers.

# The intersection of DEVELOPMENT and OPERATIONS

Check out our new  
ADMIN DevOps  
corner!

[www.admin-magazine.com/DevOps](http://www.admin-magazine.com/DevOps)

SPONSORED BY



Linux  
Professional  
Institute

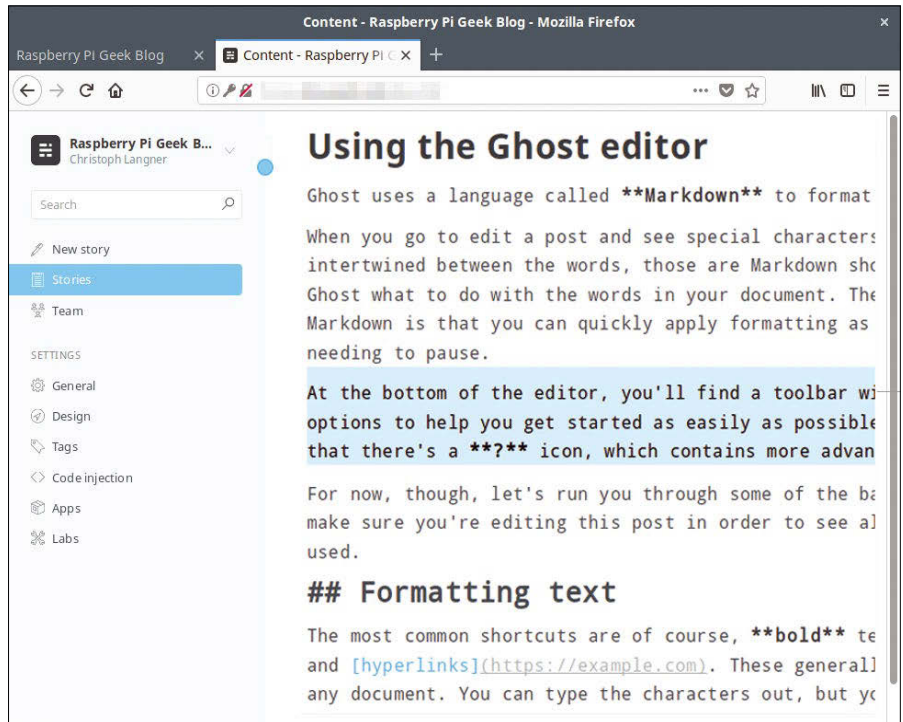


Figure 2: Ghost offers many of the standard CMS features.

and instead uses a simple Markdown [9] editor. The syntax is easy to learn. For example, asterisks format *italic* and **bold** text. Lists, links, and code blocks can be created just as easily. Buttons on the lower edge and a side-by-side preview help with the first steps.

Further support for writing articles can be obtained in the desktop client [10] and the smartphone app for Android [11]. If you don't like the look and feel of the blog, Ghost Marketplace [12] offers free and commercial themes. Pages like Ghost for Beginners [13] list more skins. In the future, the blog system will also

be expandable with plugins (known as Ghost Apps) [14].

## Conclusions

All in all, Ghost offers many features found in the CMS mainstream (Figure 2). However, the newcomer still lacks a kind of dashboard that provides information about page access and page views, as well as other essential blog features such as a way to leave comments. However, these functions can be retrofitted, for example, with Google Analytics and Disqus or, if you prefer open source tools, with Matomo (formerly Piwik) [15] and Isso [16]. ■■■

## Info

- [1] CMSs for websites: [https://w3techs.com/technologies/overview/content\\_management/all](https://w3techs.com/technologies/overview/content_management/all)
- [2] Ghost: <https://ghost.org/>
- [3] Project Ghost: <https://john.onolan.org/project-ghost>
- [4] Ghost Kickstarter: [https://www.kickstarter.com/projects/johnonolan/ghost-just-a-blogging-platform?ref=nav\\_search&result=project&term=ghost%20blog](https://www.kickstarter.com/projects/johnonolan/ghost-just-a-blogging-platform?ref=nav_search&result=project&term=ghost%20blog)
- [5] Ghost(Pro): <https://ghost.org/pricing/>
- [6] Node.js: <https://nodejs.org>
- [7] Installation instructions: <https://docs.ghost.org/v1.0.0/docs/install>
- [8] Nodemailer: <https://github.com/nodemailer/nodemailer/tree/0.7>
- [9] Ghost Markdown: <https://help.ghost.org/article/4-markdown-guide>
- [10] Desktop client: <https://ghost.org/downloads/>
- [11] Android client: <https://play.google.com/store/apps/details?id=org.ghost.android>
- [12] Marketplace: <https://marketplace.ghost.org>
- [13] Ghost for Beginners: <https://www.ghostforbeginners.com/themes>
- [14] Apps: <https://apps.ghost.org>
- [15] Matomo/Piwik: <https://matomo.org>
- [16] Isso: <https://posativ.org/isso>

The sys admin's daily grind: sshuttle

# Great Shuttle Service

When he doesn't want to deal with OpenVPN version conflicts or congestion control problems during TCP tunneling, Charly catches a ride on sshuttle. *By Charly Kühnast*

In untrustworthy networks, I let OpenVPN tunnel my laptop. There are certainly alternatives, and I would like to present a particularly simple one: sshuttle [1]. As the name suggests, the tool relies on SSH. The tunnel's endpoint is a leased root server, just like with OpenVPN. Sshuttle is very frugal. It only needs SSH access with user privileges on the server; root privileges are not necessary. Additionally, Python must be installed on the server – that's it.

This is because sshuttle loads and executes the required Python code on the server after the SSH connection is established. It also avoids version conflicts between server and client software. The following command is all it takes to set up the tunnel:

```
sudo sshuttle -r <User>@<Server>:<Port> 0/0
```

You can leave out the port number if it is the SSH standard port 22. The 0/0 means that Linux should direct all connections into the tunnel. However, this means that I cannot reach other devices in the local network. To keep the local LAN still visible, I define it as an exception using the -x parameter:

```
sudo sshuttle -r --dns <User>@<Server> 0/0 -x 192.168.2.0/24
```

--dns is included here. This means that DNS queries also run through the tunnel, which does not happen automatically. This is sshuttle's Achilles heel: It only transports TCP; ICMP and UDP do not pass through the tunnel, apart from DNS.

## Congestion Alert

Whereas other VPN technologies work at packet level and rely on TUN/TAP de-

vices, sshuttle works at session level. It assembles the TCP stream locally, multiplexes it over the SSH connection, while keeping the status, and splits it into packets again on the destination side.

This avoids the TCP-over-TCP problem which plagues other tools such as OpenVPN: TCP has an overload control (congestion control). The protocol defines a performance limit on the basis of dropped packets. If you tunnel TCP over TCP, you lose congestion control for the inner connection, which can lead to bizarre error patterns. Sshuttle is immune to the problem.

Verbose parameters can help if you do need to troubleshoot. Figure 1 shows a connection setup with -v. With the verbose option, sshuttle is very long-winded, so I recommend redirecting the output to a file that can be evaluated in peace. My conclusions: Sshuttle is an excellent and simple VPN for people who can do without UDP and ICMP. ■■■

## Info

[1] sshuttle:  
<https://github.com/apenwarr/sshuttle>

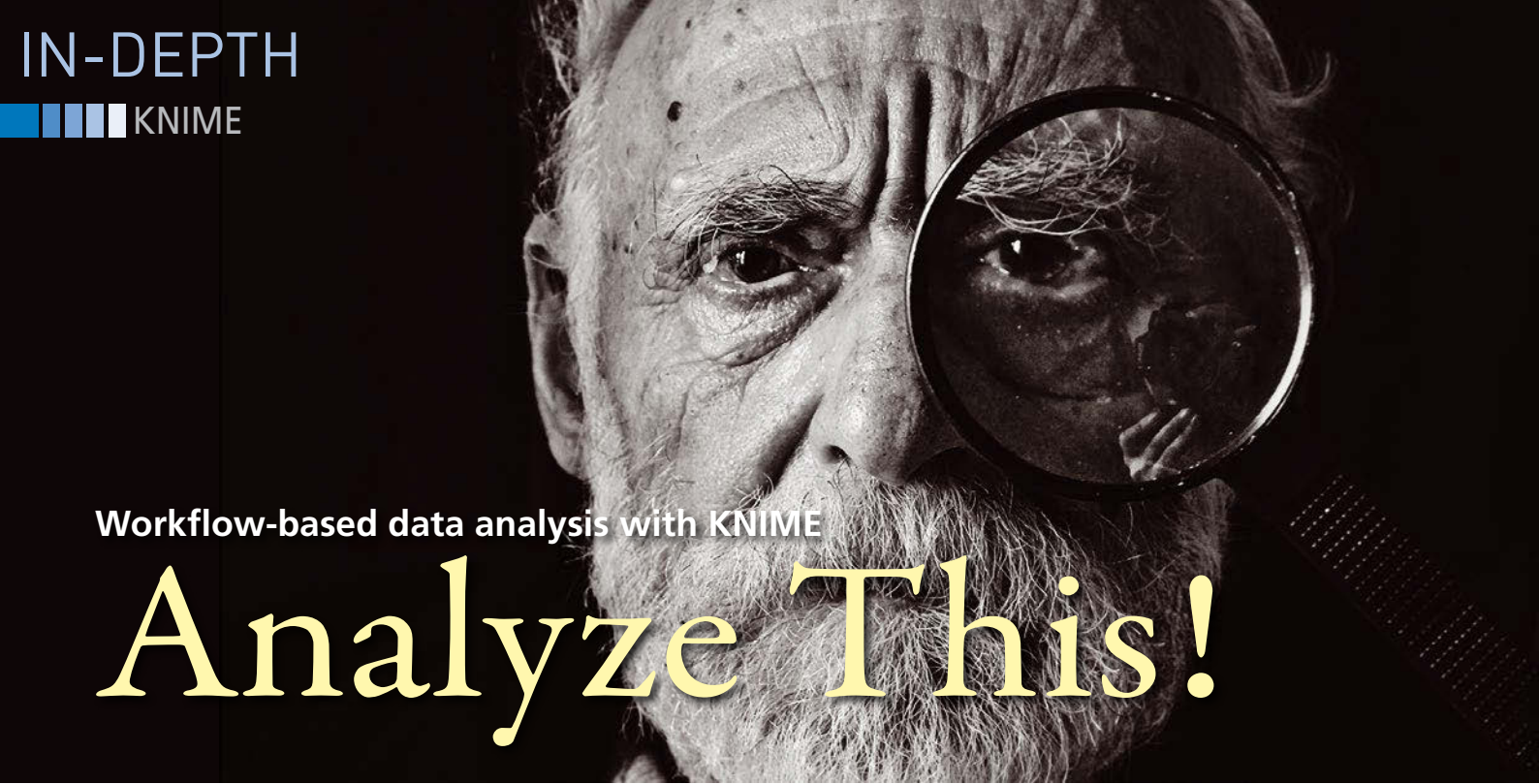
```
charly@x1:~$ sudo sshuttle --dns -v -r sshuttle@kuehnast.com 0/0
[sudo] password for charly:
Starting sshuttle proxy:
firewall manager: Starting firewall with Python version 3.5.2
firewall manager: ready method name nat.
IPv6 enabled: False
UDP enabled: False
DNS enabled: True
TCP redirector listening on ('127.0.0.1', 12300).
DNS listening on ('127.0.0.1', 12300).
Starting client with Python version 3.5.2
c : connecting to server...
sshuttle@kuehnast.com's password:
Starting server with Python version 2.7.6
s : latency control setting = True
s : available routes:
s : 2/18 8.0.2/32
s : 2/18 8.0.0/24
s : 2/89 238.82.0/24
c : Connected.
firewall manager: setting up.
>> iptables -t nat -A sshuttle-12300
>> iptables -t nat -F sshuttle-12300
>> iptables -t nat -I OUTPUT 1 -j sshuttle-12300
>> iptables -t nat -I PREROUTING 1 -j sshuttle-12300
c : iptables -t nat -A sshuttle-12300 -j RETURN --dest 127.0.0.0/8 -p tcp
c : iptables -t nat -A sshuttle-12300 -j REDIRECT --dest 0.0.0.0/8 -p tcp --to-ports 12300 -m ttl ! --ttl 42
c : iptables -t nat -A sshuttle-12300 -j REDIRECT --dest 127.0.1.1/32 -p udp --sport 53 --to-ports 12300 -m ttl ! --ttl 42
c : Accept TCP: 10.0.0.238:82842 -> 52.218.17.12:443
c : DNS request from ('127.0.0.1', 68949) to None: 33 bytes
c : Accept TCP: 10.0.0.238:47669 -> 37.129.191.252:443.
```

Figure 1: Sshuttle builds a VPN for a server. Because of -v, the messages are more extensive than without.

## 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.





Workflow-based data analysis with KNIME

# Analyze This!

They say data is “the new oil,” but all that data you collect is only valuable if it leads to new insights. An open source analysis tool called KNIME lets you analyze data through graphical workflows – without the need for programming or complex spreadsheet manipulation.

By Alexander Fillbrunn and Martin Horn

**D**ata analysts like to use flexible scripting languages such as R or Python that come with large ecosystems of libraries and extensions. But many users don’t want to have to write and debug their own custom programs just to analyze data.

Visual workflows offer a different approach. You can use visual workflows to break down the analysis processes into modular, sequential steps. Each step is symbolized by a graphic element called a *node*. Each node performs an action, which might be a calculation, a formatting function, or another step related to data analysis and manipulation. By linking the nodes on the screen, users can create workflows for complex investigations of the data – without producing any code.

Visual workflows are the central element of the KNIME Analytics Platform. In

### The Authors

**Alexander Fillbrunn** is a PhD student at the Department of Bioinformatics and Information Mining at the University of Konstanz. He is particularly interested in the development of machine learning algorithms.

**Martin Horn** is a postdoc at the Department of Bioinformatics and Information Mining at the University of Konstanz, where he studies data analysis – of course using KNIME.

the KNIME environment, a workflow is a graph with nodes showing a series of sequential steps for processing and analyzing the data. The user defines a pathway for the data by connecting the output of one node to the input of another node. A type system ensures that you can only connect compatible output and input. Real programming code is only necessary if you want to integrate KNIME with R or Python – or if you want to develop your own modules.

KNIME, which is pronounced “*nime*,” was originally known as the Konstanz Information Miner; it began in 2006 at the Department of Bioinformatics and Data Mining at the University of Konstanz under the direction of Prof. Michael Berthold. The basic idea was to make data analysis easily accessible and affordable for users from different disciplines. The developers, therefore, turned their creation into an open source project, paying particular attention to making the tool extensible and user-friendly.

KNIME Analytics Platform is written in Java and is based on Eclipse and Open Services Gateway Initiative (OSGI) technology. The latest version (version 3.5.1 at the time this article was written) is available at the project website [1], where you will also find other introductory ma-

terials, including blogs, videos, and sample workflows.

The KNIME Analytics Platform is open source; anyone can download and use it free of charge. Start the installer or unpack the archive, and KNIME is ready to go. However, you might wish to add some KNIME extensions, which are the source for many additional nodes. To add an extension, go to *File | Install KNIME Extensions*, check the list for the desired extension, and follow the instructions.

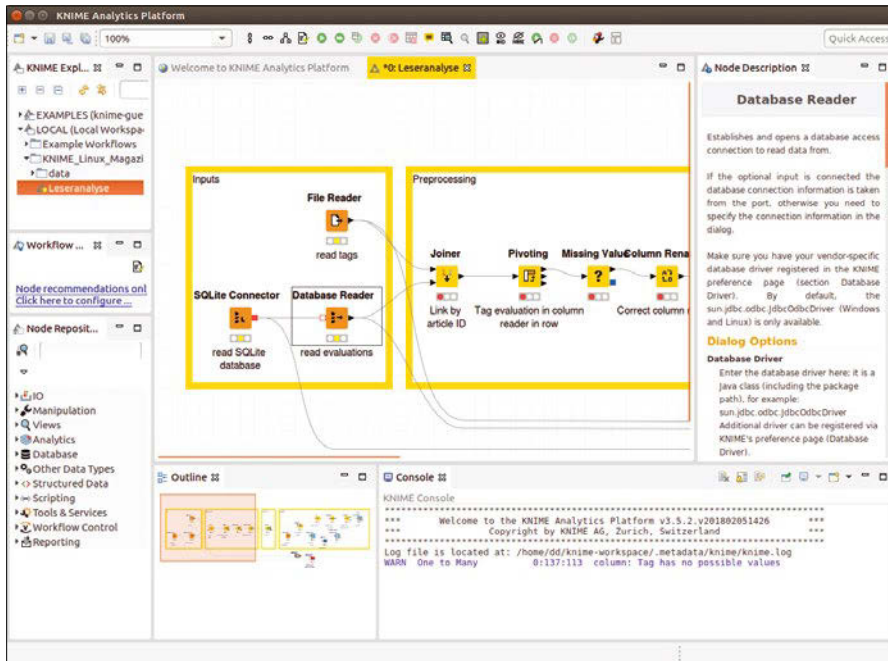
### Looking Around

Figure 1 shows the KNIME user interface. The KNIME Explorer in the top-left corner of the workspace provides an overview of the workflows. Click the *View* menu and select *Workflow Coach* to access the Workflow Coach, which offers suggestions on building a workflow. The Node Repository (bottom left) lists the nodes of all installed extensions.

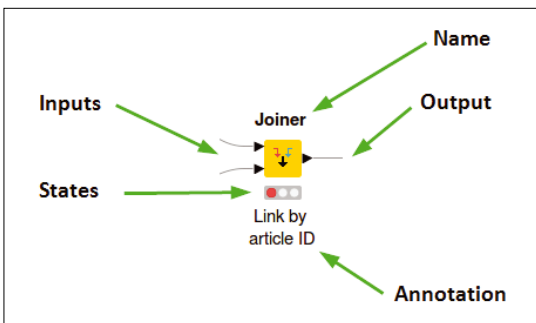
The nodes are the building blocks of any KNIME workflow, and the vast library of available nodes gives KNIME its versatility and power. KNIME nodes perform tasks such as:

- data access
- data manipulation
- visualization
- analytics
- reporting

Lead Image Photo by mari lezhava on Unsplash



**Figure 1:** The KNIME user interface, in which users compose workflows using drag & drop.



**Figure 2:** Anatomy of a KNIME node.

- flow control
- scripting
- big data

The node description in the window on the right of the user interface gives the user the necessary documentation on the function and use of a node. In the middle is the workflow editor, where users string the nodes together to develop the actual workflow. During the course of developing a workflow, the user connects, configures, and executes the KNIME nodes individually and collectively (Figure 2).

Once a node has been executed, which is indicated by a green traffic light symbol below the node, you can display the resulting data as a table, bar chart, or other format. If the traffic light symbol is red, the node is not yet configured. Yellow means the node is ready for execution.

Nodes can have between zero and an arbitrary number of inputs and outputs (ports). The ports' shape and color indi-

cate what kind of data the node needs or outputs at a particular input or output (a black triangle indicates a table).

### Sample Scenario

The best way to get to know KNIME is to work through an example workflow (Figure 3). The editors of a fictitious online magazine would like to know more about their readers' preferences. They have gathered

**Table 1:** Evaluation from a SQLite Database

Reader ID	Article ID	Evaluation
Reader 1	Article 11	1
Reader 93	Article 31	3
Reader 45	Article 3	4
...		

some data from a rating system that gives readers the opportunity to rate magazine articles with up to five stars. Each article is also classified in at least one of the five categories: Hardware, Software, Development, Security, and Internet.

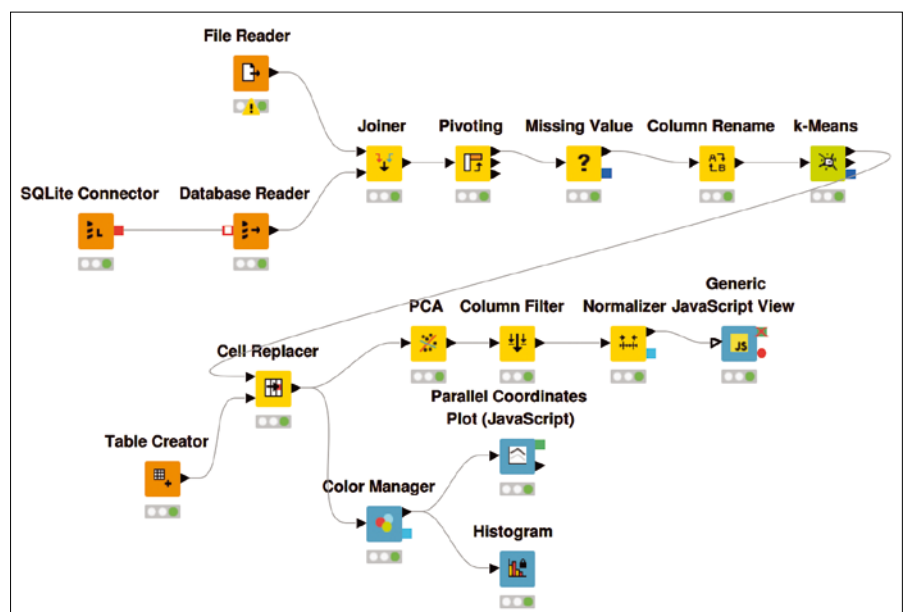
The editors hope to draw conclusions from the ratings on reader preferences and to identify groups of readers with common interests. They then want to suggest further articles to each reader that match their interests.

The starting points for the analysis are a SQLite database with the article ratings (Table 1) and a CSV file with the classification of the categories for the articles. Both data sources are therefore available in a table format.

### Loading the Data

The first step of any data analysis is loading the required data (the red nodes in Figure 3). For KNIME, the data can come from a variety of sources, such as text files, documents, databases, or web services.

Once loaded in KNIME, it doesn't matter where the data came from, because the software always converts the data



**Figure 3:** The sample workflow, which analyzes reader preferences for a publisher.

into an internal format. In this case, I need to load data from an SQLite database and a CSV file. To help you access databases, KNIME has several nodes to assist with data input, including the Database Reader node, which delivers the results of an SQL query as a KNIME table.

The node works with different database types and receives the connection information via its inbox. You can prefix an SQLite connector, which is configured with the connection information and forwards it to other nodes.

The easiest way for KNIME to load text files is to use the File Reader. This node automatically attempts to guess the file format, including column separators and row length, and displays a preview of the table in its configuration dialog.

### Opening a Connection

The next step is to relate the imported data to each other. If you are familiar with databases, you know that the best way to do this is with a join. A join consists of one or more predicates that determine which rows belong together in two tables. The rows that belong together are then combined in an output table.

This example needs to combine the two tables based on the article ID. In this case, the join predicate determines that ratings for an article must be linked to the categories to which the same article is assigned. The node that KNIME uses for this is Joiner. Its output table is shown in Figure 4. Like its database counterpart, Joiner supports various modes, including inner, outer, left, and right joins. The user can also select which columns are to be copied from which table into the output.

### Pivoting

The next step is to create a kind of fingerprint for each reader, describing the

S Article...	S Tag	S Reader ID	I Rating
Article142	Security	Reader171	2
Article175	Software	Reader234	2
Article9	Software	Reader178	2
Article17	Software	Reader219	3
Article50	Security	Reader141	4
Article97	Development	Reader302	4
Article70	Development	Reader22	2
Article98	Internet	Reader294	3

Figure 4: Output table for the Joiner node.

reader's preferences by calculating the reader's average rating for the articles in each section.

The result is a table in which each reader becomes a row and each category becomes a column. The cells of the intersection points display the reader's average rating for articles in each category (Figure 5).

This transformation is handled by the pivot node. The pivot node dialog may prove intimidating at first glance, since it requires information on the group, pivot, and aggregation columns in three separate tabs. The values in the group columns determine which rows need to be combined into a single row – that is, all those that have the same values in these columns. For the sample workflow, the reader ID is a group column, the category is the pivot column, and the aggregation column contains the ratings that an average function averages.

If a reader has not rated a single article for a category, the cell in the pivoted table does not contain a value. KNIME indicates this with a red question mark. You can replace missing values using the Missing Value node. In the example, it makes sense to assume that the reader does not like a category in which they have never rated an article. Instead of the missing value, the table will use a zero.

It is often the case that a node produces a table with long or incomprehensible column names. For example, the pivot node from the previous category creates columns in the format "Category + Rating." The suffix, which is the same in each column, is not important. Two important KNIME nodes assist with renaming columns. Column Rename lets you rename each column name explicitly; Column Rename (Regex) lets you change the column names using regular expressions.

### Cluster Analysis

Now that the data is transformed

S Reader ID	D Development	D Hardware	D Internet	D Security	D Software
Reader63	4	3.2	2.857	2.889	4.375
Reader270	3.857	3.714	3.625	3.727	3.857
Reader56	4.111	2.889	2.286	2.75	4.286
Reader148	4.4	4.571	3.167	4.2	3
Reader82	3.2	2.75	4.375	3.444	4.111
Reader202	4.1	2.833	4.3	4.1	4.143
Reader122	3.083	4.4	3.091	4.4	2.2

Figure 5: Pivoting results.

into a workable format, you can begin to group readers according to their preferences. Readers with similar preferences should end up in the same group. Cluster analysis is a useful technique for this kind of problem, and KNIME has different nodes to support it, including k-means, fuzzy c-means, hierarchical agglomerative clustering, or k-medoids.

What all these cluster algorithms have in common is that users have to decide for themselves how many groups they want to receive. They also all need a way to calculate the similarity between two datapoints. In the previous category, a table row was created for each reader, containing five numeric values – one for each category.

The workflow for the example uses the k-means node, which is quite fast and generally produces good results. The algorithm works with different distance measurements – in KNIME it is implemented with the Euclidean distance (see the "Computing the Distance" box). The k-means node in KNIME, which executes the algorithm, appends another column to the input table that remembers the group in which the datapoint lies.

Cluster analysis reveals that readers who give a high rating to articles in the categories Hardware and Security usually lie in cluster 0, while readers from cluster 1 prefer the categories Internet and Software, and cluster 2 prefers Development and Software. Once you know the results, it is possible to give the clusters meaningful names. Due to their preferences, for example, readers in Cluster 0 can be addressed as administrators, those in Cluster 1 as power users, and those in Cluster 2 as developers. These new names are added using the Cell Replacer node.

### Visualization: Histogram

A histogram, provided by the histogram node, provides an overview of how many readers lie in which group. The histogram node can be connected directly to the Cell Replacer from the previous step, but the whole thing looks a bit colorless. The Color



## Computing the Distance

Let each reader be a five-component vector consisting of the ratings in the five categories. The following formula then computes the distance between the two readers  $x$  and  $y$ :

$$\sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_5 - y_5)^2}$$

The k-means algorithm initiates the following steps to find homogeneous groups:

1. Initialize the cluster centers with random datapoints.
2. Compute the nearest cluster center for each datapoint.
3. Compute the center for each group of datapoints assigned to a center in Step 2.
4. Update the cluster centers with the newly found centers.
5. If the centers remain unchanged, stop the computations; otherwise go to step 2.

Manager node lets you assign a color – based on the value of a column – to the rows of a table. If the same column is also used for the x-axis of the histogram, KNIME automatically takes over the color for the bars (Figure 6).

## Visualizing Multiple Dimensions

Visualizations are available to display data and also show whether the discovered clusters have any significance. In the past years, many visualization methods were implemented in KNIME with the help of JavaScript and the D3.js framework [2].

These visualizations are available in the KNIME JavaScript Views extension,

which is where you will also find the Parallel Coordinates visualization. Parallel Coordinates represents the properties of the data with parallel y-axes (Figure 7).

D3.js, the JavaScript library on which most KNIME JavaScript visualizations are based, is one of the most widely used libraries for creating interactive data visualizations in the browser. However, the KNIME user can only use some of its possibilities.

For all cases that KNIME does not yet cover, use the Generic Javascript View. Configuring this node means you can enter arbitrary JavaScript and CSS code to compute a colorful image from a table. The code executed by the node has access to the node table and the browser Document Object Model (DOM) and can generate HTML and SVG elements based on the data.

For example, you can use the Generic Javascript View to create a Voronoi diagram (Figure 8), which visualizes the clustered reader groups in 2D. To ensure that the data is in a format suitable for visualization, you must first reduce the number of dimensions. Up to now, the example has used five dimensions per reader (one for each section), but I will now break this down into two using the principal component analysis as calculated by the PCA (Principal Component Analysis) node. This type of transformation reduces the dimensionality, but at the same time tries to keep the variance in the data as complete as possible so that as little information as possible is lost.

By this point, a proper data analysis workflow has already been created: from importing the data, through transformation and grouping, to visualizing the

results. All this helps to gain interesting insights into the raw data. In a further step, it is now possible to identify for each reader the articles that a reader has not yet read, but which could also be interesting for the reader because of their interests. A web application can then suggest these articles to the reader.

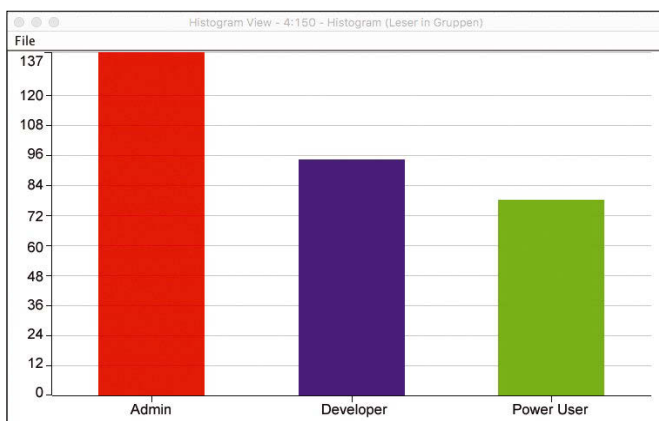
## Keeping Track

With each further step, the workflow threatens to become more complex and confusing. To ensure that it remains comprehensible, it is a good idea to encapsulate individual parts in modules to conceal the complexity. This encapsulation is made possible using so-called meta nodes. Meta nodes also let you group the sections of the workflow using meaningful names. Figure 9 shows a possible restructuring of the workflow using meta nodes.

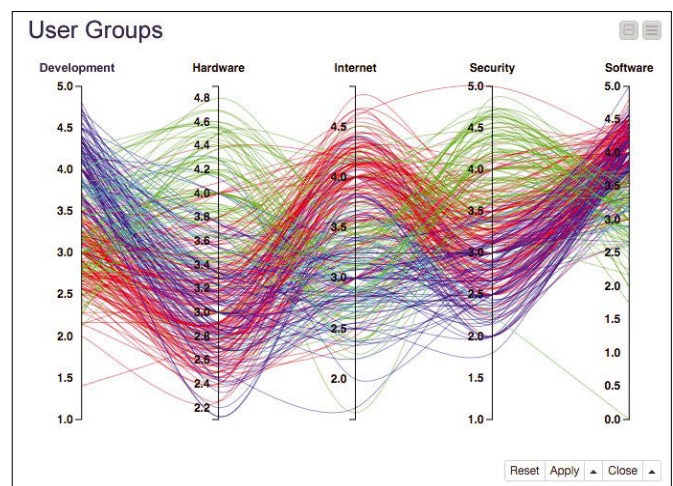
## Recommended Reading

The point of the exercise is to recommend articles that are closest to the reader's preferences. For example, if the reader is particularly interested in hardware and security, it would be a good idea to suggest articles from these categories – or even articles that are in both categories at the same time. The current workflow has already explored the extent to which a certain reader has a preference for each category, and the result is available in the form of a vector (Figure 5).

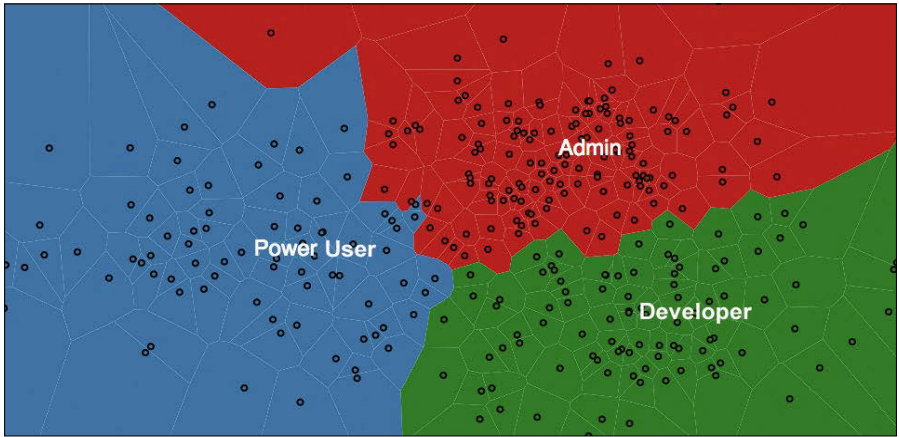
You can create a very similar vector for each article, where the columns of those categories contain a 1, to which the article is assigned. Category columns without connection to the article, on the other



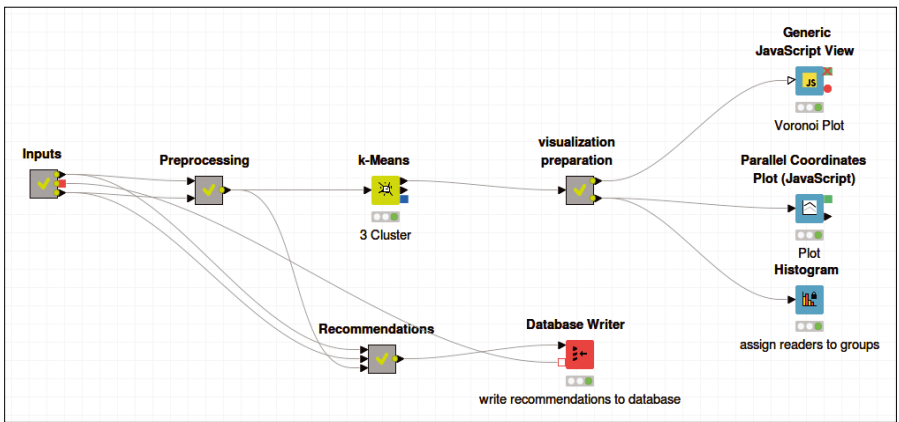
**Figure 6:** Histogram of the groups of readers as determined by cluster analysis.



**Figure 7:** A parallel coordinate chart of reader preferences. Each color represents a group of readers.



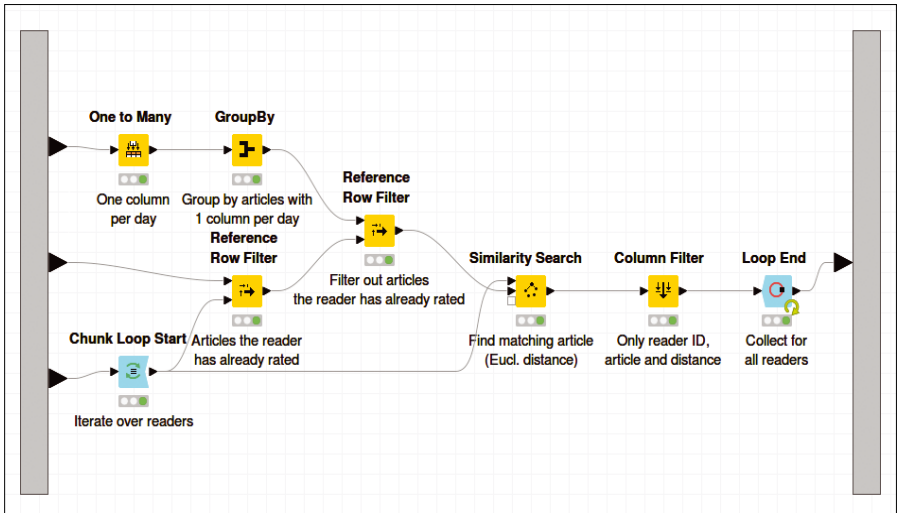
**Figure 8:** A Voronoi diagram of the readers created with the Generic Javascript View.



**Figure 9:** The sample workflow from Figure 3 with parts of the workflow summarized as meta nodes.

S	Article...	Development	Security	Hardware	Software	Internet
Article83	1	0	0	0	0	0
Article7	0	0	1	0	0	0
Article49	0	0	0	0	0	1
Article71	0	1	0	0	0	0
Article25	0	0	0	0	0	1
Article35	1	0	0	0	0	0
Article69	1	0	0	0	0	0
Article81	0	1	0	0	0	0

**Figure 10:** Output table of the one-to-many node.



**Figure 11:** The content of the meta node for article recommendations.

**Table 2: Categories Table**

Article ID	Category
Article 11	Hardware
Article 11	Software
Article 31	Development
...	

hand, contain a 0. The One-to-Many node makes this possible: The transformation of the article-heading assignments (Table 2) into a representation by a binary vector per article (Figure 10).

Once the two vector types have been created, the Similarity Search node can simply determine a distance between the vector of a reader (which represents its preferences) and the vector of an article (which indicates to which columns it is assigned). The smaller the distance, the more the reader's preference corresponds to the categories in which the article is classified.

From all articles that a certain reader has not yet read (see Row Reference Filter node), it is now easy to determine the article that has the shortest distance to the reader's preferences. This article is finally recommended for reading.

A loop (consisting of the Chunk Loop Start and Loop End nodes) corresponds to a For loop across all rows of the table. The loop determines the smallest distance of the vectors for each reader. The overall result with one article recommendation per reader (Figure 11) could then be written back to a database with the help of the Database Writer node.

### Conclusion

This article only covers a fraction of the nearly 2,000 nodes available for KNIME. Other exciting KNIME features include flow variables, the Workflow Coach, streaming, nodes for processing texts, and Deep Learning capabilities.

You can get an idea of the KNIME Analytics Platform by downloading and testing the software. Check out the documentation [3] and node guide [4] at the KNIME website for more on working with KNIME, or bring your questions to the KNIME Forum [5]. ■■■

### Info

- [1] KNIME: <https://www.knime.com>
- [2] D3.js framework: <https://d3js.org>
- [3] KNIME Documentation: <https://www.knime.com/documentation>
- [4] KNIME Node Guide: <https://www.knime.com/nodeguide>
- [5] KNIME Forum: <https://www.knime.com/forum>

# Now Appearing on APPLE NEWSSTAND

## New age convenience...

Our inspired IT insights  
are only a tap away.

Look for us on  
Apple Newsstand  
and the iTunes store.

Download  
a FREE issue of  
each publication  
now!



## Manipulating Binary Data with Bash

# Bit Trip

Bash is known for admin utilities and text manipulation tools, but the venerable command shell included with most Linux systems also has some powerful commands for manipulating binary data. *By Andy Carlson*

One of the most versatile scripting environments available on Linux is the Bash shell. The core functionality of Bash includes many mechanisms for tasks such as string processing, mathematical computation, data I/O, and process management. When you couple Bash with the countless command-line utilities available for everything from image processing to virtual machine (VM) management, you have a very powerful scripting platform.

One thing that Bash is not generally known for is its ability to process data at the bit level; however, the Bash shell contains several powerful commands that allow you to manipulate and edit binary data. This article describes some of these binary commands and shows them at work in some practical situations.

## Viewing and Converting Data

Two tools you can use to represent data in hexadecimal and binary format are `hexdump` and `xxd`. The `hexdump` utility provides many options for outputting hexadecimal data. Although `xxd` lacks some of the options available with `hexdump`, it does have one key feature `hexdump` lacks: In addition to letting you output hexadecimal values, `xxd` also has the ability to convert a hexadecimal string into binary data. The following command:

```
command> echo -n "hello" | xxd -p
output> 68656c6c66
```

outputs the binary values of the ASCII string “hello” as an ASCII string of hexadecimal values (refer to Table 1).

In the following command:

```
command> echo -n "6162" | xxd -p -r
output> ab
```

The hexadecimal string “6162” is converted into what the `xxd` man page calls a “mail-safe ASCII representation” of the binary data. Because hexadecimal 61 and 62 correspond to ASCII characters “a” and “b,” respectively, the binary data is expressed in the form of the string “ab.”

Bash makes it easy to build these commands into functions. The following functions, `pack` and `unpack` (named because of their similarity to the PHP `pack` function) use the preceding commands to convert a hexadecimal string to binary and convert binary to hex.

```
pack() {
    echo -n "$1" | xxd -p -r
}

unpack() {
    echo -n "$1" | xxd -p
}
```

All the examples so far have used the `-p` switch to specify that the command will

use plain hexdump style, representing each 8-bit byte as a two-digit hexadecimal number. To output data as a binary string (containing 1s and 0s), you need to use different switches. The following function returns a binary string representation of

**Table 1: ASCII Lowercase Alphabet**

Char	Decimal	Hexadecimal
a	97	61
b	98	62
c	99	63
d	100	64
e	101	65
f	102	66
g	103	67
h	104	68
i	105	69
j	106	6A
k	107	6B
l	108	6C
m	109	6D
n	110	6E
o	111	6F
p	112	70
q	113	71
r	114	72
s	115	73
t	116	74
u	117	75
v	118	76
w	119	77
x	120	78
y	121	79
z	122	7A

binary data, along with a sample command and output:

```
tobin() {
    echo -n "$1" | xxd -b -g0 |
    awk '{ printf("%s", $2) }'
}

command> tobin ab
output> 0110000101100010
```

The two-character string “ab” is converted into a binary string containing two 8-bit values: 01100001 and 01100010 (97 and 98), corresponding to the base 10 values for characters “a” and “b.”

## Transforming Binary Data

Bash also offers some commands for transforming binary data using mathematical and logical operations. This process is called arithmetic expansion. In the following example:

```
command> echo $(( 4*5 ))
output> 20
```

the output of the echo command is the value calculated by the statement enclosed in `$(( and ))`.

Another tool for performing mathematical operations is `bc`. `bc` is a command-line calculator that can read statements from standard input. For example:

```
command> echo "4*5" | bc
output> 20
```

Note that the preceding operation is performed in base 10. `bc` does not provide native functionality for arithmetic operations on hexadecimal numbers, so you need to convert the numbers to base 10 and then perform the operation. For instance, if you want to perform simple addition and subtraction operations against hexadecimal numbers, use a function like the following:

```
hexadd() {
    echo "obase=16;ibase=A;
    $((16#$1)+$2" | bc
}

command> hexadd A 2
output> C
```

The `hexadd` function provides the ability to add or subtract (add a negative number)

to a valid hexadecimal number in a single command. The function pipes a string of commands into `bc`. The first command, `obase=16`, sets the base in which data will be output (in this case, hexadecimal or base 16). The second command, `ibase=A`, sets the base used to read input data. The input base is set up `A`, which corresponds to base 10. The third and final command is an addition statement comprised of an arithmetic expansion and the second function argument. The arithmetic expansion uses the `#` operator to convert the number following it from the base specified before the `#` to base 10. In the command, the hexadecimal value for “A” is converted to decimal 10 using arithmetic expansion, added to 2 using `bc`, and then converted back into hexadecimal using `bc`.

You can also perform bitwise operations on the data, including AND, OR, XOR, and shift. The AND operation (using the `&` operator) returns the bits shared between the two numbers. The OR operation (using the `|` operator) returns all bits used by either of the two numbers. The XOR operation (using the `^` operator) returns all bits that are unique to one of the numbers.

In Listing 1, the first command performs a binary AND on 3 (or 0011 in binary) and 6 (or 0110 in binary). Because the only shared bit is the second bit, 2 (or 0010 in binary), the output for the command is 2. The second command performs an OR operation against the same two numbers: 3 and 6. Between the two numbers, the lower three bits are used, so 7 (or 0111 in binary) is returned. In the third example an XOR is performed against the same two numbers. The second and fourth bits are the same between the two numbers, however, the third and first bit are different, therefore 5 (or 0101 in binary), is returned.

The other bitwise operations is the shift operation, which shifts the bits in one direction or another (right or left). Listing 2 shows functions for shifting the bits shifting right (using the `>>` operator) and left (using the `<<` operator).

The first command in Listing 2 shifts the bits in 4 (0100) left by two spaces, returning a value of 16 (10000). The second command shifts the bits in 16 (10000) to the right four spaces, returning a value of 1 (0001). As you might have noticed, for each bit space shifted to the left, the number is multiplied by 2,

### Listing 1: Bitwise Operations

```
01 logicand() {
02   echo $((($1&$2))
03 }
04
05 logicor() {
06   echo $((($1|$2))
07 }
08
09 logicxor() {
10   echo $((($1^$2))
11 }
12
13 command> logicand 3 6
14 output> 2
15 command> logicor 3 6
16 output> 7
17 command> logicxor 3 6
18 output> 5
```

and for each bit space shifted to the right, the number is divided by 2.

## Examples

Switching from binary to hexadecimal, and moving bits around to change an A into a C, is certainly interesting, but does this capability have any uses in the real world? The following examples offer some hints for how you could use these tools in practical ways.

### ASCII to Unicode

In some situations, you might need to convert an ASCII character into Unicode. ASCII is an 8-bit character set, whereas Unicode starts at a 16-bit length. Converting from ASCII to Unicode might seem complicated, but it is actually quite simple thanks to the backward compatibility built into the Unicode standard. To convert ASCII to Unicode, you just need to prepend the value of 0 onto each

### Listing 2: Shifting Bits

```
01 shiftr() {
02   echo $((($1>>$2))
03 }
04
05 shiftl() {
06   echo $((($1<<$2))
07 }
08
09 command> shiftl 4 2
10 output> 16
11 command> shiftr 16 4
12 output> 1
```

ASCII character, thus making it a 16-bit character (see Listing 3).

In Listing 3, the output of the command appears to show no noticeable change. To get a better view of the binary data behind this text, pipe the output into `xxd`:

```
command> ascii2unicode jello | xxd
output> 0000000: 006a 0065 006c 006e
006c 006f      .j.e.l.l.o
```

As you can see, the ASCII values have been prepended with “00,” which converts them to 16-bit Unicode characters. Take a closer look at Listing 3 to see what’s happening: The output of the `echo` statement is piped into the `sed` statement, which places each character of the output on a separate line. The `awk` command reads the input from the `sed` command

### Listing 3: ASCII to Unicode

```
01 ascii2unicode() {
02   echo "$1" | sed 's/\(.\)/\1\n/g' | awk '/^.{}/ { printf("%c%c",0,$0) }'
03 }
04
05 command> ascii2unicode jello
06 output> jello
```

### Listing 4: URL Encoding and Decoding

```
01 urlencode() {
02   echo -n "$1" | xxd -p | tr -d '\n' | sed 's/\(.\)/%\1/g'
03 }
04
05 urldecode() {
06   tr -d '%' <<< "$1" | xxd -r -p
07 }
08
09 command> urlencode name
10 output> %6e%61%6d%65
11 command> urldecode %64%6f%6e%65
12 output> done
```

### Listing 5: Converting a Subnet Mask

```
01 subnetcalc() {
02   echo -n "$1" | \
03   awk 'BEGIN { FS="."; printf("obase=2;ibase=A;") } { printf("%s;%s;%s;%s\n", $1,$2,$3,$4) }' | \
04   bc | sed 's/^0$/0000000/g;s/\(.\)/\1\n/g' | \
05   awk 'BEGIN { ht = 0; nt = 0; }
06       /[01]/ { if ($0=="1") nt++; if ($0=="0") ht++; }
07       END { printf("Network bits: %s\nHost bits: %s\nHost IP Count: %d\n",nt,ht,2^ht); }'
08 }
command> subnetcalc 255.255.192.0
output> Network bits: 18
output> Host bits: 14
output> Host IP Count: 16384
```

line-by-line, and when the line contains a single character, it prints the character prepended by the character value “0”.

### URL Encoding and Decoding

Hexadecimal data is something you see every day, but it often goes unnoticed. When data is passed as a query string in a URL, it may be encoded using special formatting. This formatting consists of a percent sign followed by the hexadecimal value of an ASCII character. For example, the URL encoded string of “%61%62%63,” when decoded, becomes “abc.” Listing 4 shows a function for performing URL encoding and decoding.

The function in Listing 4 uses the standard functionality of `xxd`. When encoding a string, the output of `xxd` is split into 1-byte chunks and prepended with a

“%” by the `sed` command. When decoding, all percent signs are stripped and the output is piped into `xxd` to revert the hexadecimal string to ASCII.

### Calculating IP Subnets

On an IP network, the subnet mask specifies how many bits of the IP address will be dedicated to the network ID and how many will be used for the host ID. The size of the host ID address space will tell you how many host IP addresses are available. Listing 5 shows how to convert the subnet mask to a binary string and determine the host ID count.

The output of the `echo` statement is fed into the `awk` statement. This first `awk` command will generate the statement that is piped into the following `bc` command. The statement will include `ibase`, `obase`, and each individual octet of the subnet mask. Once `bc` evaluates the statements, it returns four lines: one line for each octet. The following `sed` statement finds lines containing only “0” and extends them to 8-bits of zeros. The `sed` statement also puts each bit on a line by itself. This will be necessary to properly evaluate the host bit length. The `awk` statement has three sections. The first section initializes the `ht` and `nt` variables, which store the host total bits and network total bits, respectively. The next section searches for lines containing 0 or 1. If the value is 1, the network total is incremented, and if the value is 0, the host total is incremented. The final section of the `awk` statement prints the summary data for the network, including the host and network bit counts, along with the host IP count.

### Conclusion

The versatile Bash shell offers several utilities that allow you to output, transform, and manipulate binary values. This article introduced you to the `xxd` and `bc` commands. You also learned how to add these binary commands to your custom Bash functions to build your own tools for performing practical tasks like decoding URLs and calculating subnet masks. ■■■

# IT Highlights at a Glance

The collage features several overlapping images of IT-related content:

- ADMIN HPC**: A magazine cover with the headline "Expand storage capacity on a budget!" and "Software-Defined Storage - Download a FREE special issue from ADMIN HPC".
- ADMIN Update - Hottest Links**: A webpage snippet with a "Further Reading" section listing articles like "Auditing Docker Containers in a DevOps Environment" and "Permissions Stored on Windows Vulnerable".
- SUSE Expert Days 2018**: A green banner with the text "Coming to a city near you." and "Linux Update: Exploring the World of Linux".
- Linux Update**: A blue banner with the text "Exploring the World of Linux" and "Featured Articles" including "Putting LibreOffice in the Cloud" and "CICP Illuminates Serverless Vision".
- ADMIN Update**: A red and white webpage snippet with a "Most Read Article" section titled "Windows Subsystem for Linux".
- ADMIN HPC Trends**: A small magazine cover with the headline "Expand storage capacity on a budget!".
- ISC High Performance**: A red and white banner for "HPC & Machine Learning Tutorials, Conference & Workshops" held from June 24-28, 2018 in Frankfurt, Germany.
- Securing Your PoC**: A small magazine cover with a blue and white design.

Too busy to wade through press releases and chatty tech news sites? Let us deliver the most relevant news, technical articles, and tool tips – straight to your Inbox. Subscribe today for our excellent newsletters:

ADMIN HPC • ADMIN Update • Linux Update  
and keep your finger on the pulse of the IT industry.

Admin and HPC: [www.admin-magazine.com/newsletter](http://www.admin-magazine.com/newsletter)  
Linux Update: [www.linux-magazine.com/newsletter](http://www.linux-magazine.com/newsletter)



# MakerSpace

A countdown counter with the MAX7221 and a seven-segment display

3, 2, 1 ... Go!

Build a countdown counter with a Raspberry Pi and some electronics, and you can count down the time to any event.

By Martin Mohr

**D**ays until Christmas, time until retirement: We count the days to many events in life. A countdown counter can help make the time to wait fly by and increase anticipation. In this article, I show you how to use the Raspberry Pi as a control unit to build a chic countdown counter based on an LED segment display.

To ensure that the countdown can be seen from a distance easily, a large

seven-segment Kingbright SC08-11SRWA [1] display (20.32mm/0.8 inch high) is used to display the remaining days to an event. The LEDs use a common cathode, which makes the display compatible with a MAX7221 display driver.

The combination of parts used here requires the use of a soldering iron. Completely assembled modules that match the capabilities of the MAX7221 are available, but they have considerably

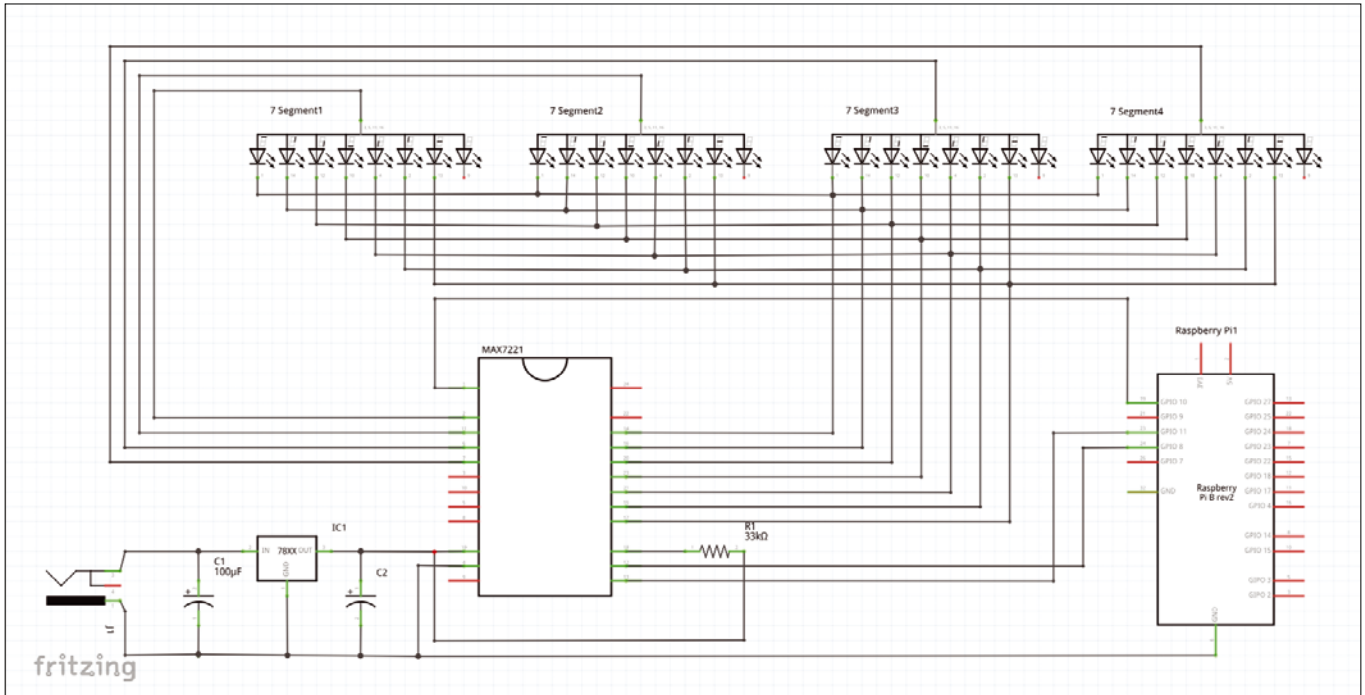
**Table 1: MAX7221: Register Assignment**

Register	Function	Description
0x00	No-op	Functionless. In the case of several SPI blocks on one bus, it is used to address only one block. All others receive a no-op.
0x01	Digit0	Digit 0 of the display.
0x02	Digit1	Digit 1 of the display.
0x03	Digit2	Digit 2 of the display.
0x04	Digit3	Digit 3 of the display.
0x05	Digit4	Digit 4 of the display.
0x06	Digit5	Digit 5 of the display.
0x07	Digit6	Digit 6 of the display.
0x08	Digit7	Digit 7 of the display.
0x09	Decode mode	Each bit corresponds to one digit of the display (bit 0 = digit 0, etc.). If it is set, the digit works as a segment display; the lower 4 bits of the digit registers are then interpreted as a BCD number. If it is deleted, each segment of the digit can be controlled as a single LED.
0x0A	Intensity	The lower 4 bits of the register control the brightness of the LEDs.
0x0B	Scan limit	The lower 3 bits of the register control which digits of the display are used (0x00 = only digit 0; 0x01 = digits 0 and 1; etc.).
0x0C	Shutdown	Bit 0 = 1: Normal mode, all displays and functions active. Bit 0 = 0: Shutdown mode, all displays off, no response to commands.
0x0F	Display test	Lights all LEDs.

## Author

Born in the era of magnetic core ring accumulators and rotary dial phones, Martin Mohr developed a love for everything that flashes at an early age. After studying computer science, the qualified electrician developed Java applications. His old passion for electronics was revived by the Raspberry Pi.





**Figure 1:** The Fritzing circuit diagram of the countdown project.

smaller displays of (typically) eight digits. Four digits should be enough for a day countdown (i.e., 9,999 days or more than 27 years). If you really want to count down for a longer period of time, simply add an additional segment to the setup.

**MAX7221 Display Driver**

In the MAX7221 display driver, an external resistor controls the current for the LEDs. I’ll shoot for 33kohm, which results in a current of just under 20mA for one segment of the display. Otherwise, the MAX7221 does not require external wiring. Internally, the display stores the information to be displayed in 8x8-bit RAM.

Table 1 is an overview of the MAX7221 registers and their functions; a data sheet [2] provides further information. The MAX7221 is addressed through the SPI interface. Many microcontrollers are suitable for this purpose – I concentrate on the Raspberry Pi.

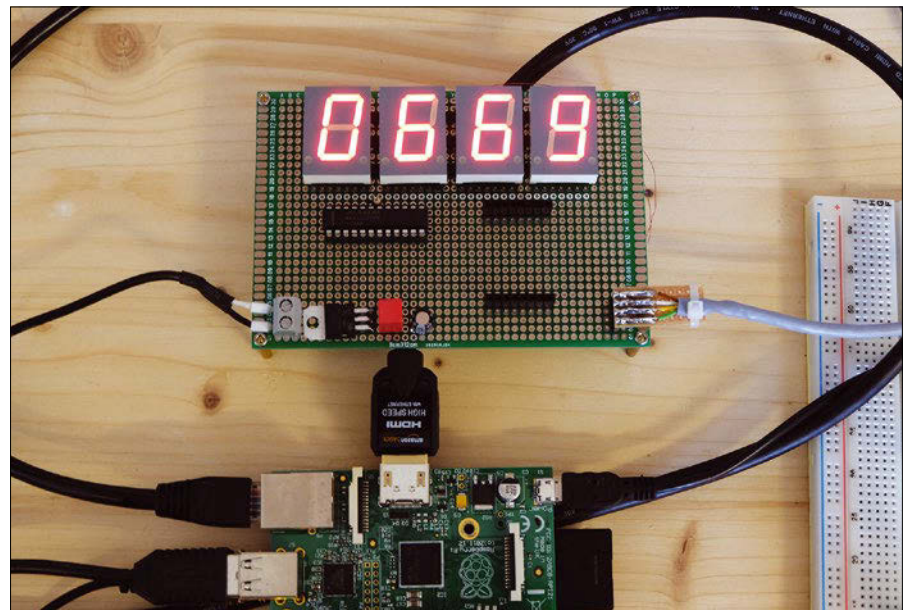
**Circuit Diagram**

In the countdown counter circuit diagram (Figure 1), the MAX7221 is the central component. Connected to the four seven-segment displays, it only requires one external component: resistor R1. The complete circuit operates at 3.3V, which is supplied by a type LD1117v33 voltage regulator by STMicroelectronics (input voltage range = 4.3-15V).

The setup is thus quite independent of the voltage source (e.g., a USB power supply, car battery, etc.). Two capacitors buffer fluctuations in the input voltage and counteract the oscil-

**Table 2: Components**

No.	Name	Source	Price/Unit (EUR)
4	SC08-11SRWA 7-segment display	reichelt.de	1.15
1	MAX7221CNG 8-digit LED display driver	reichelt.de	5.99
1	PCB prototype 8x12	aliexpress.com	~1
	Various small parts	Craft kit	2
<b>Total cost ~EUR20</b>			



**Figure 2:** (Top) Countdown counter on a Euroboard; (bottom) Raspberry Pi.

lation of the voltage regulator. A header plug joins the connectors of the SPI interface to the Raspberry Pi. Figure 2 shows the complete setup; all components are listed in Table 2.

## Program

The experimental setup is based on a Raspberry Pi 1 with a current Raspbian image, but the steps described here also work with any other Raspberry Pi model.

## Cron

The Unix cron service allows commands and scripts to be executed automatically at certain times. The `crontab` command-line tool is used for configuration. Use `crontab -l` to see the current settings for the logged in user. The `-e` switch puts Crontab in edit mode. An entry contains, separated by white space, the specifications for minute, hour, day of the month, month, and weekday, as well as the command to be executed at the specified time.

The first five parameters control execution time. Numbers stand for times or dates and an asterisk for any given minute, hour, or day. If necessary, you can also use a backslash to define multiple values (e.g., `*/2` for every second minute). As far as timing is concerned, it is important to ensure that one call can be processed before the next initiates; otherwise, the system will become overloaded over time.

The following are some typical cron examples:

```
* * * * * /home/user/script # every minute
0 */2 * * * /home/user/script # at the start of every other hour
0 0 * * * /home/user/script # every day at midnight
30 18 * * * /home/user/script # every day at 6:30pm
* * 1 * * /home/user/script # every minute on the first day of the month
```

Further detailed information can be found on the cron man page. Useful pages on the web, such as at Corntab [3], can help you to compose cron lines. You need to make sure that cron executes scripts with the rights of its own user account but does not use environment variables. To do this, specify all paths in full and do not use aliases.

### Listing 1: counter.c

```
01 #include <fcntl.h>
02 #include <sys/ioctl.h>
03 #include <linux/spi/spidev.h>
04 #include <inttypes.h>
05 #include <stdio.h>
06 #include <time.h>
07
08 static const char *device = "/dev/spidev0.0";
09 static uint32_t speed = 500000;
10 int handle, diff;
11 time_t ts_now, ts_target;
12 struct tm target;
13
14 void write_register(int handle, uint8_t regi, uint8_t value) {
15     uint8_t tx[] = { regi, value };
16     write(handle, tx, 2);
17 }
18
19 int main() {
20     if ((handle = open(device, O_RDWR)) < 0) {
21         perror("SPI Device Error");
22         return 1;
23     }
24     if (ioctl(handle, SPI_IOC_RD_MAX_SPEED_HZ, &speed) < 0) {
25         perror("SPI Speed Error");
26         return 1;
27     }
28     // MAX7221-Setup
29     write_register(handle, 0x0C, 0x01);
30     write_register(handle, 0x09, 0xFF);
31     write_register(handle, 0x0F, 0x00);
32     write_register(handle, 0x0B, 0x03);
33     write_register(handle, 0x0A, 0x0F);
34     ts_now = time(NULL);
35     target.tm_mday=1;
36     target.tm_mon=4; // Month 1 (Jan = 0)
37     target.tm_year=119; // Year - 1900
38     target.tm_sec=59;
39     target.tm_min=59;
40     target.tm_hour=23;
41     ts_target=mktime(&target);
42     diff=(ts_target-ts_now)/(60*60*24);
43     write_register(handle, 0x04, (diff%10000)/1000);
44     write_register(handle, 0x03, (diff%1000)/100);
45     write_register(handle, 0x02, (diff%100)/10);
46     write_register(handle, 0x01, (diff%10));
47 }
```

First, activate the SPI interface with `raspi-config` in *5 Interfacing Options | P4 SPI*. After the next reboot, the `/dev/spi-dev0.0` Raspberry Pi device directory contains the SPI device, which can be used as a character-oriented device under Unix.

Listing 1 shows the `counter.c` example program, which loads all the libraries used and defines the necessary variables. The `write_register()` function (lines 14-17) encapsulates the `write()` C function and arranges the parameters accordingly. The `main()` function (line 19 to end of code) establishes the connection to the SPI device and sets the speed of the SPI interface. It then initializes the MAX7221 to use digits 0 to 3 in decoder mode and sets the brightness to the maximum.

The target date structure (line 12) generates the timestamp of the target date. This structure has some peculiar characteristics that can cause a great deal of trouble: Days count from 1 and months from 0, and 1900 must be deducted from the year. The values for time of day can be written one to one into the structure.

The remaining lines of the program calculate the difference between the

current and target dates and write it to the corresponding memory locations of the MAX7221. By transferring the start of the program to a cronjob (see the “Cron” box), you can start the program once a day, calculate the current display value, and transfer it to the display driver. Save the program and compile it with `gcc` (GNU C compiler):

```
$ gcc counter.c -o counter
$ ./counter
```

The last line runs the program.

## Conclusions

A close look at the MAX7221 reveals that the device is ideal for controlling LEDs: It supports up to 64 individual diodes, the design does not require any series resistors, and the brightness of the LEDs can be adjusted through the module. By the way, if the eagerly awaited event is very far into the future, you should consider finding a more energy-efficient supplier of SPI signals than the Raspberry Pi, such as an ESP8266. ■■■

## Info

- [1] SC08-11SRWA single-digit numeric display data sheet: <http://www.kingbrightusa.com/images/catalog/SPEC/SC08-11SRWA.pdf>
- [2] MAX7221 data sheet: <https://datasheets.maximintegrated.com/en/ds/MAX7219-MAX7221.pdf>
- [3] Corntab: <http://corntab.com>

# Shop the Shop

# shop.linuxnewmedia.com

## Missed an issue?

You're in luck.

Most back issues are still available. Order now before they're gone!

[shop.linuxnewmedia.com](http://shop.linuxnewmedia.com)

**GET IT NOW!**

SAVE TIME ON DELIVERY WITH OUR ALTERNATIVE PDF EDITIONS





# MakerSpace

Reinventing the keyboard

## Custom Keyboard

The Keyboardio Model 01 is a highly-customizable open hardware keyboard. *By Bruce Byfield*

For a while, the Keyboardio [1] Model 01 seemed destined to never ship. Although its 2015 crowdfunding campaign raised over five times its goal of \$120,000, production was repeatedly thwarted by problems with manufacturers and quality control – all of which is recorded in painstaking detail on the company blog [2], and which should become required reading for any would-be hardware vendor. Yet despite everything, in November 2017, the Model 01 keyboard finally began shipping. Al-

though shipping is still seriously backlogged, the Model 01 (Figure 1) turns out to be worth waiting for, both as a high-end ergonomic keyboard and as one of the most customizable pieces of open hardware ever produced.

The Model 01 is a mechanical keyboard built with Matias keyswitches [3] that are not only longer-lasting and easier to repair than the usual cheap membrane keyboards, but require less pressure to activate than standard keyswitches. The minimalist keyboard of 64 keys ships with three layers – that is, key bindings that can be activated while pressing a command key, just as the default lowercase keys produce uppercase letters when the Shift key is pressed. With the Model 01, pressing the Num key activates the number pad and arrow keys, while pressing the Fn key activates the 12 function keys, Delete, and square and curly brackets. The default layers are illustrated on a laminated sheet that ships with the Model 01.

Keys are divided between two slabs of maple hardwood. Each of the keys is individually sculpted and shaped to help guide fingers from one key to the other. To reduce finger movement, rows of character keys are arranged in curving key banks and columns of keys on a straight diagonal, instead of having straight rows and columns staggered to the right, as on a standard keyboard. Command keys are to the left and right of the letter keys, or else arranged in an



**Figure 1:** The Model 01 straight out of the box.

arc of four at the bottom of the keyboard that are pressed by the thumbs.

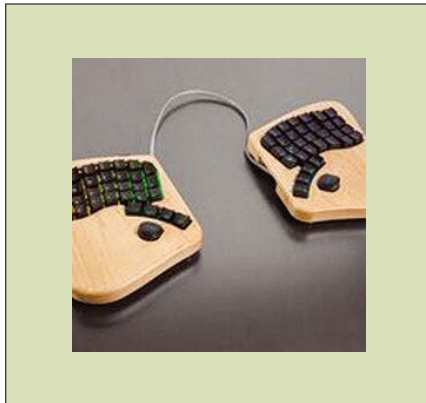
Besides the Fn key, the command keys include the replacement for a Windows key, the LED key for toggling through installed backlight settings, the Prog key for flashing the firmware, and the Any key, which originated in the old joke about procedure instructions to “press any key,” but became a blank key that can be programmed as desired. The keyboard ships without a Cap lock button, but, since the Shift key is accessed by the thumb, many users can continue to type while holding it down.

### The Advantages of Ergonomics

The arrangement and shapes of the Model 01’s keys, as well as the choice of keyswitches, are designed to reduce the typing stress. The result is that typing on a Model 01 is a different experience from typing on most keyboards – especially from the cheap membrane keyboards sold in most computer stores. As the Quick Start Guide notes, learning to type at your accustomed speed on a Model 01 can take as long as several weeks (for me, it was about five days). If my own experience is any indication, a new user can expect to start with a typing technique not far removed from hunt and peck, then gain about half their usual speed until they become too tired to remember their new reflexes, and after that to slowly and steadily regain their usual cruising speed.

This process can be trying, but it has the advantage of suggesting how you might customize the keyboard layout to your individual advantage (see below). For example, I find myself constantly overreaching when I type an A or a Z and hitting the PgUp or PgDn keys beyond them. Therefore, I decided to remap the two keys I was hitting by mistake so that they copy and paste instead.

Moreover, contrary to what cynics might assume, in the case of Keyboardio’s Model 01, the word “ergonomic” is more than a marketing term (often used the way that “organic” or “gluten-free” might be used in the supermarket). My own experience is that, after a few days of stiffness as finger muscles are used in unfamiliar ways, the twinges and disablements of repetitive stress injuries are reduced to short-lived ghosts of their former selves.



**Figure 2:** Depending on the connecting cable, the Model 01 halves can be placed together or widely separated.

This ergonomic advantage is no doubt due to the fact that the Model 01 began as a highly personal project of Keyboardio’s technical partner Jesse Vincent. Those who regularly spend over eight hours a day at a keyboard can especially appreciate the advantages. Although it almost seems beside the point, the arrangement of keys on an arc and their mounting on maple also makes the Model 01 the most stylish keyboard since those produced by custom keyboard maker Datamancer [4] – although after the industry-standard ugliness, the aesthetics are a welcome change.

### Hardware Customizations

When ordering a Model 01, you can choose one programmed by default for the standard QWERTY layout, or alternative layouts such as Dvorak, Colemak, Malt, or Workman. You also have the choice of keycaps – including Runic or blank – and can buy extra sets as well. In addition, you have the choice of the default quiet keyswitches or the loud keyswitches that some typists prefer for their audio feedback.

Once your order arrives, you can experiment with several different layouts. A Model 01 is split into two halves (Figure 2). Each half contains a ATmega32u4 Arduino micro-controller and is connected to the



**Figure 3:** The stands and centerplate allow the keyboard to be positioned in several different ways.

other by an RJ45 cable – a short one for those who want to place the halves close together, or a longer one for users who prefer to separate them. The halves can be joined by a center bar or kept separate so that they can be positioned individually on a flat surface.

Still another option is to place the two halves flat or to mount each on its own stand at a variety of angles (Figure 3). For ergonomic purposes, the most efficient arrangement is to tent each half, so that the inner edge is higher than the outer (Figure 4). If the centerplate is also used, the stands must be screwed into position after the centerplate is added.

Without any software editing, users can also change the keyboard backlighting by toggling the LED key (Figure 5). A dozen different default options are available, ranging from a half a dozen backlight colors, the lighting up of a pressed key, the blinking on and off of



**Figure 4:** The two sides positioned at an angle.

the entire array of keys, or turning all backlights off. If you have never used backlit keys, you may be inclined to dismiss them as frivolous, but they may help to make your typing more accurate by making the keys more visible, especially in dim light. The Model 01 uses low-draw backlights, so using them should have minimal effects on the longevity of the hardware.

Do-it-yourselfers can even open up the keyboard to adjust the microcontrollers. Unlike with other, proprietary products, user exploration of the hardware does not void the warranty. In fact, to emphasize Keyboardio's dedication to openness, the Model 01 ships with a small multi-headed screwdriver suitable for opening it up.

## Keyboard Customization

Unlike Input Club [5], another maker of open source keyboards, Keyboardio does not yet have a graphical interface for customizing keybindings. An interface called Chrysalis [6] is in development, but it is not yet officially released. Nor does the Model 01 ship with any instructions on how to modify Kaleidoscope [7], the Model 01 firmware – apparently on the assumption that those interested will know enough to find the online instructions. However, even if they do, instructions and resources are scattered over several sites.

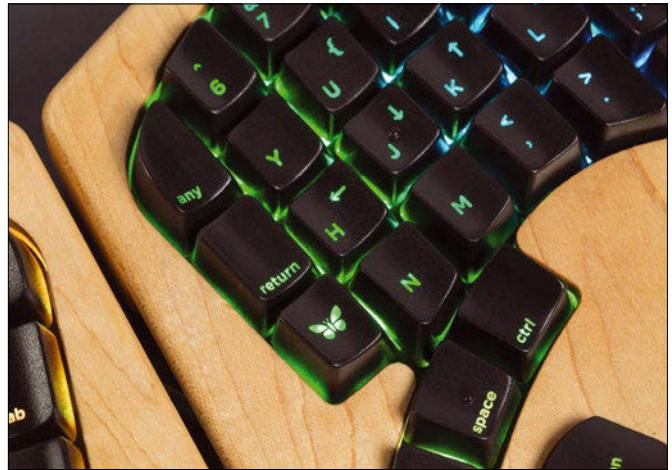
Customizing the Model 01's behavior means modifying the firmware and reinstalling or flashing the result – the identical requirement for installing updated firmware from Keyboardio. The full set of instructions and their background are too lengthy to give here, but it requires the installation of an Arduino environment and Kaleidoscope on a computer to which the keyboard can be attached [8]. Once this prerequisite is completed, users can modify keymaps, macros, and plugins, either by themselves or by downloading and pasting keymaps created by other users.

The simplest modification is editing the keymapping to change what happens when each individual key is pressed (Figure 6). Keymapping is set in the `Model01-Firmware.ino` file, which is an example of what Arduino programming calls a Sketch. A keymapping is literally that – a text-based map in which each

key is named along with what happens when that key is pressed. The default firmware defines the default layer for a QWERTY key layout and separate layers for the keybindings when the Fn and Num keys are pressed. Users can modify the default layers, enabling unused

keys, or adding new ones, such as traditional alternative layouts like Dvorak or Colemak [9]. Other popular modifications include rearranging the command keys such as Space or Esc to preferred positions. Some definitions for keys can be learned from the default file or from the online reference [10].

To a certain extent, these changes can be reinforced by repositioning the easily removable keycaps on the Model 01. However, the individually-sculpted keys may limit the repositioning, which is why additional sets of keycaps – especially blank ones – are useful.



**Figure 5:** Backlights make keys more visible.

Like keymaps, macros can also be added directly to `keymap Model01-Firmware.ino`. For macros for a specific piece of software, such as Vim, you can add a new layer. Some users have also created new LED effects as plugins, which are accessed via macros. Basic instructions for writing a macro are available online [11], and the existing macros can also be tweaked or commented out. A partial list of existing macros is available online [12]; an especially useful one is `MacrosOnTheFly`, which can record keystrokes and play them back as macros [13].

```
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_LEffectNext,
    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 6:** A default keymapping displayed in the Arduino IDE.

More complex modifications can be placed in plugins, which generally require some Arduino programming knowledge. A description of core – that is, official plugins – is available [14], as well as descriptions of unofficial ones [15].

The resources listed above are reasonably complete, but hackers should watch the Keyboardio community site [16] for other examples and instructions. There has been some talk of a central repository and manuals, but, as I write, neither exists.

All these modifications can be written in the Arduino IDE or in a text editor – after, of course, a backup is created for recovering from disastrous results. When the modifications are complete, save the revised Sketch in ~/Arduino/Model01-Firmware. From the same directory, run the command `make flash`. When the output signals that the command is ready, hold down the Prog key so that the command can bypass the microcontrollers' bootloaders before continuing. The Prog key will flash to show that it is being used. When the command is complete, unplug the keyboard, and plug it in again so that the firmware can restart.

### Rethinking the Keyboard

On the company blog, Keyboardio tells how at initial meetings, potential manufacturers suggested that the keys should be mounted on white plastic, or that the

firmware should be protected as intellectual property. However, Keyboardio remained determined to produce a quality open source product, and, despite the manufacturing delay and the chaotic modification instructions, it has generally delivered.

True, without shipped instructions and a graphical interface, the Model 01 is less accessible than it should be. It is also true that other keyboards have some of its features. But the function-

ality is there, and no other keyboard offers anything near the Model 01's combination of ergonomics, aesthetics, capacity for customization, and quality hardware.

In the Model 01, Keyboardio shows how much can still be done to improve a piece of hardware that has changed little for several decades. That is no mean accomplishment and makes Keyboardio an outstanding example of what open hardware can accomplish. ■■■

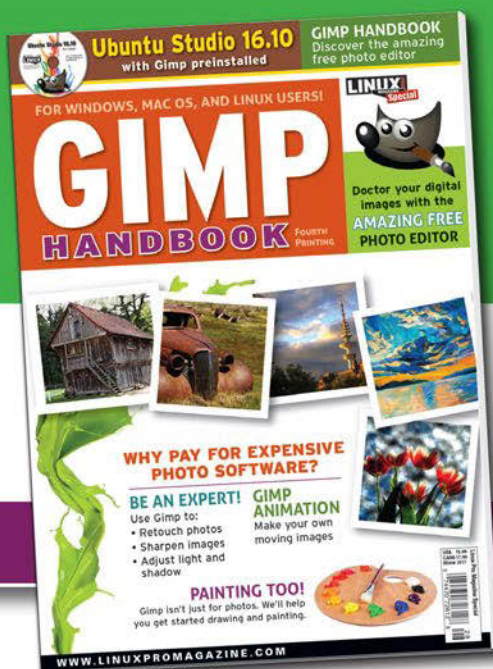
### Info

- [1] Keyboardio: <https://shop.keyboard.io>
- [2] Keyboardio blog: <http://blog.keyboard.io/>
- [3] Matias keyswitches: <http://matias.ca/switches/linear/>
- [4] Datamancer: <https://datamancer.com/>
- [5] Input Club: <https://input.club/>
- [6] Chrysalis: <https://community.keyboard.io/t/chrysalis-layout-keyboard-settings-editor/253/21>
- [7] Kaleidoscope: <https://github.com/keyboardio/Kaleidoscope>
- [8] Installing Arduino IDE and Kaleidoscope: <https://github.com/keyboardio/Kaleidoscope/wiki/Keyboardio-Model-01-Introduction>
- [9] Traditional alternative keymaps: <https://github.com/keyboardio/Kaleidoscope/wiki/Common-Alternate-Layouts>

- [10] Key definitions: [https://github.com/keyboardio/Kaleidoscope/blob/master/src/key\\_defs\\_keyboard.h](https://github.com/keyboardio/Kaleidoscope/blob/master/src/key_defs_keyboard.h)
- [11] Writing a macro: <https://github.com/keyboardio/Kaleidoscope-Macros/blob/master/README.md>
- [12] Partial list of macros: <https://github.com/algernon/Model01-sketch#installation>
- [13] MacrosOnTheFly: <https://github.com/cdisse/Keyboardio-MacrosOnTheFly>
- [14] Official plugins: <https://github.com/keyboardio/Kaleidoscope/wiki/Core-Plugins>
- [15] Unofficial plugins: <https://community.keyboard.io/c/programming/Discuss-Kaleidoscope-Plugins-one-thread-per-plugin>
- [16] Keyboardio community site: <https://community.keyboard.io>

Shop the Shop

[shop.linuxnewmedia.com](http://shop.linuxnewmedia.com)



# GIMP HANDBOOK

- Fix your digital photos
- Create animations
- Build posters, signs, and logos



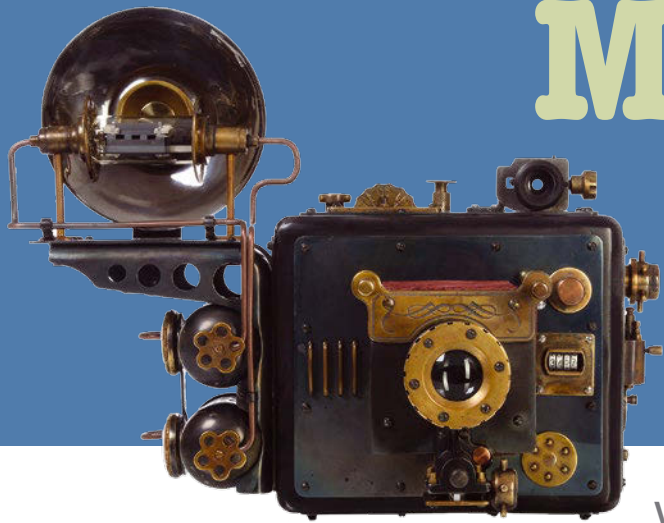
Order online: [shop.linuxnewmedia.com/specials](http://shop.linuxnewmedia.com/specials)

FOR WINDOWS, MAC OS, AND LINUX USERS!



# MakerSpace

## Advertising with the Raspberry Pi Show Talent



Video sequences or slide shows promote sales. With a Raspberry Pi and the Raspberry Slideshow distribution, you can capture the attention you need. *By Erik Bärwaldt*

**T**hanks to its active community and incremental improvements, the Raspberry Pi has long since surpassed the point of purely experimental applications. In the meantime, the miniature PC has made itself useful in the commercial environment, for example as a controller for industrial machines.

For this article, the Raspberry Pi comes into play where previously expensive TV sets with VCRs or, in recent times, individual computers with displays have been used for advertising purposes. With Raspberry Slideshow [1] and the tiny computer, you can bring multimedia content to the screen professionally and at a very low price.

Thanks to intelligent software, the Rasp Pi can easily supplement advertising columns and billboards. You can use the Raspbian-derivative Raspberry Slideshow to play a digital photo album or fade in video sequences, which can be mixed if necessary.

The operating system is available in two functionally different versions: In contrast to the Donors' Build, the free version does not include an option for customizing the configuration over SSH. In the free version, then, certain manual modifications to configuration files can only be carried out with the

help of a Linux PC, from which you can edit the corresponding files directly on the SD card.

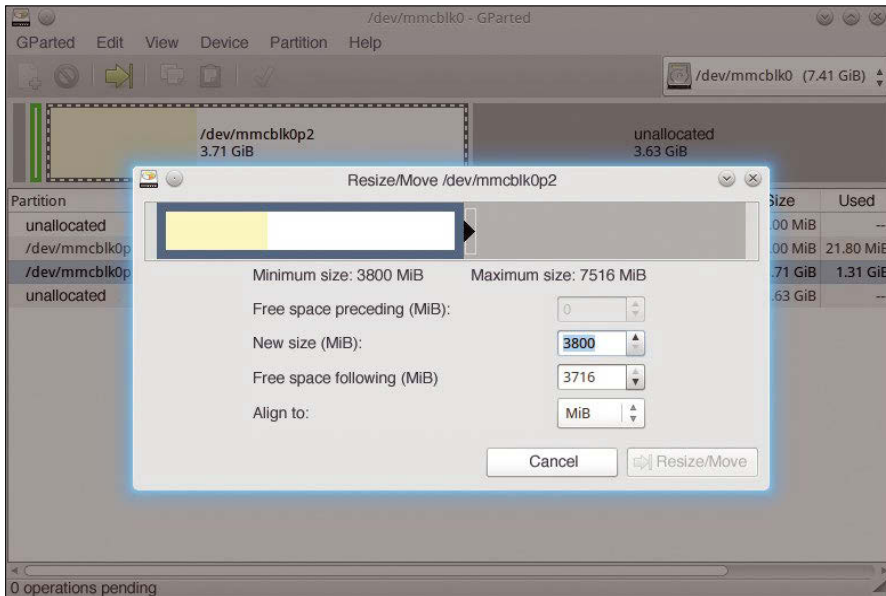
Additionally, the freely available version only lets you play back multimedia content stored on a USB memory stick or on the Internet. Different server lists and automatic rotation of the images on the basis of existing Exif data in presentations can only be handled by the Donors' Build [2], which is available for EUR16.90.

### Getting Started

Raspberry Slideshow is available as a 580MB ZIP archive on the SourceForge site. After unpacking the archive, transfer the extracted image to an SD card at least 4GB in size, although a higher capacity memory card is recommended with an expanded system partition (Figure 1).

The software copies the contents from a USB stick to the card so that files can be played back automatically, even without an external memory device. If space runs out when copying content to the SD card, the software cancels the process without an error message. As a result, the system will then only display the data in the slide show that has actually made it onto the card.





**Figure 1:** With GParted, you can increase the storage capacity for the Raspberry Slideshow system.

After installing the image on the SD card, start your Rasp Pi with the Slideshow system. The integration of external data sources into the network requires wired LAN access if an Raspberry Pi 2 (RPi2) is used, because the system does not detect external wireless dongles in the standard version. Even the adapters directly supported by Raspbian are not integrated during startup.

The operating system plays a video show for demonstration purposes: the northern Italian city of Verona at night. The calculator displays the show in full-screen mode and does not allow you to control it with the keyboard or mouse.

## Content

To view content in the appropriate order, add an ext4 or (V)FAT filesystem to a USB stick. Create a single partition only, and then place the desired videos or pictures directly in the main directory. Raspberry Slideshow is not able to read content from a directory hierarchy.

Connect the prepared data carrier to the Rasp Pi and reboot. After powering up, the content automatically appears on the screen, with the Rasp Pi's HDMI port also transmitting audio signals. The restriction on sources in the freely available version of Raspberry Slideshow reduces the benefit somewhat: You are allowed to play a maximum of three files of mixed content with the standard version. This restriction is removed only in the commercial version.

Raspberry Slideshow uses the command-line program `Fbi` to display a photo slide show (JPG, PNG, BMP, and GIF) and `OMXPlayer` to play videos (MP4 and AVI).

If the system temporarily or continuously displays a yellow flash symbol at the top right of the screen after power-up, the power supply is not performing well enough, which could cause slide shows to stop prematurely or the system to transfer video files incompletely to the SD card, resulting in a black screen.

Make sure that the power supply of the RPi2 outputs at least 1,200mA, and

for the Raspberry Pi 3 (RPi3) at least 2,500mA. Problems often arise when buying cheap power supplies from the Far East that come directly from the manufacturer: Often they do not comply with current EU standards and fail to deliver the promised performance.

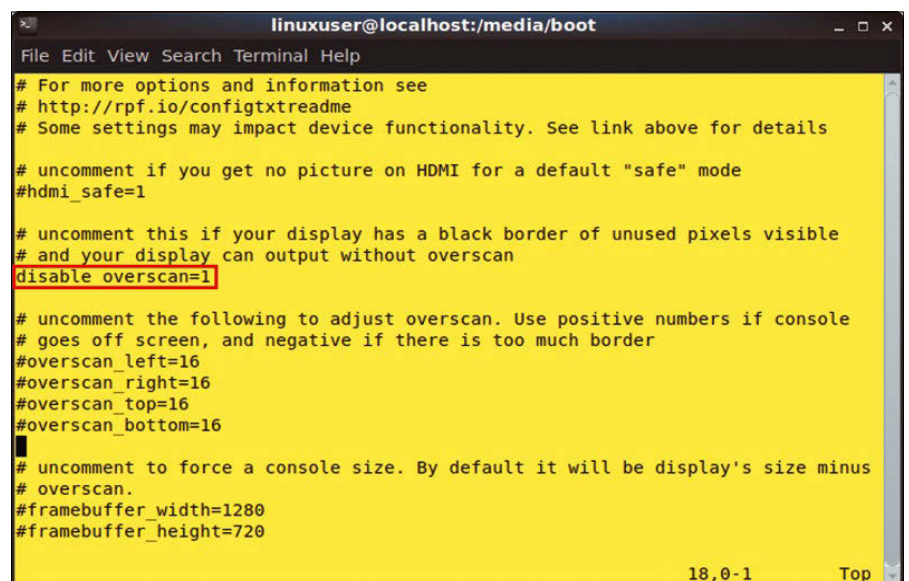
## Configuration

Different form factors between photos and screens often result in black edges on the sides and occasionally on the upper and lower edges. A similar phenomenon often occurs when playing back videos that were produced with a 4:3 aspect ratio.

To eliminate such unsightly edges as much as possible, expand the display area by mounting the SD memory card with the boot partition on a Linux PC and editing the `/boot/config.txt` file. You need to enable the `disable_overscan=1` line, which is commented out, simply by removing the preceding hashtag (Figure 2). After a reboot, the system displays the videos and images in full-screen mode, eliminating the annoying black borders when the image conforms to the monitor.

Another problem arises when playing back large image collections: The `Fbi` player uses only a relatively small buffer, causing jerky transitions when changing images. To prevent this, edit the `/etc/rs.conf` file located in the root partition of the SD card.

This file controls the parameters for starting the two playback programs. For



**Figure 2:** Modifying the `/boot/config.txt` file gives you superior image reproduction.

```

linuxuser@localhost:/media/202638e1-4ce4-45df-9a00-ad725c2537bb/etc
File Edit View Search Terminal Help
SOURCE_FOLDER="/m/usb"

TRANSITION_TIME_S=10
BLEND_TIME_MS=600
IMAGE_PLAYER="fbi -autozoom -norandom -timeout $TRANSITION_TIME_S -noverbose
-blend $BLEND_TIME_MS -vt 1 -readahead -cachemem 600 -once -list"
IMAGE_PLAYER_IMAGES_ONLY="fbi -autozoom -norandom -timeout $TRANSITION_TIME_S
-noverbose -blend $BLEND_TIME_MS -vt 1 -readahead -cachemem 600 -list"
EXIF_ROTATE="n"

VIDEO_PLAYER="omxplayer -t off -r -b"

DEBUG="n"

1,1 All

```

**Figure 3:** The `/etc/rs.conf` file manages the player configuration.

Fbi, the command sequences contain the `cachemem` parameter, which specifies the maximum size of the buffer used. Here, you need to increase the specified value to avoid jerky transitions.

In this file, you can also modify the parameters for cross-fade effects and display time. You can also rotate images or play them in a random order (Figure 3). Detailed instructions can be found in the software's documentation [3].

### Off the Web

If you want to load content from a web server instead of copying it to the USB stick, instruct the system to retrieve the data directly from the Internet or LAN. To do so, create a simple text file named `remote-urls.txt` on the stick, in which you enter one URL per line that refers to the content. You can use both web and FTP servers as a source.

Store this text file in the main directory of the disk and restart Raspberry Slideshow. It now retrieves the images and videos from the web and plays them back locally. You may mix local and remote content in one session.

The commercial version of the software also lets you retrieve content for a slide show from a Samba server and play back individual content on a remote server. Examples and useful documentation can be found on the project page [4].

### Hands On

In our lab, Raspberry Slideshow was pretty impressive. However, the restrictions of the free version – the lack of SSH support and a limitation to only three files – greatly reduce the utility value. On the up side, it is noticeable that the software (and this is not stated in the documentation) works without connected input devices.

Raspberry Slideshow makes it easy to show videos or picture collections in an endless loop, such as in shop windows or at an exhibition. Moreover, the Rasp Pi can be attached discreetly to the back of the display, which avoids unnecessary cable clutter.

Another interesting feature is automatic image rotation, which is only available in the commercial version: It uses the Exif data of the photos, guaranteeing that all images appear in the correct orientation. This version also enables the use of the integrated wireless chip in the RPi3, which allows remote configuration (e.g., from a notebook over SSH). Content can also be transferred to the Rasp Pi in this way, making it easier to update displays.

On a 3K monitor connected to the RPi2 (Figure 4), some jerkiness was observed in high-resolution photos. However, this effect was eliminated by adjusting the buffer memory.

### Conclusions

Raspberry Slideshow turns the single-board computer into a full-fledged advertising system for trade fairs, exhibitions, and demonstrations. Flexible handling of the sources, various possibilities for customization, and simple configuration in only two text files are all appealing. The full range of functions is only available with the commercial option, but the price you invest is money well spent. ■■■

### Info

- [1] Download: <https://sourceforge.net/projects/raspberrypictureframe/files/>
- [2] Purchase a license: <http://www.binaryemotions.com/raspberry-slideshow-download/>
- [3] FAQ: <http://www.binaryemotions.com/raspberry-slideshow-faq/>
- [4] Examples: <http://www.binaryemotions.com/raspberry-slideshow-download/>



**Figure 4:** The Rasp Pi also looks good with 3K and 4K displays.

Shop the Shop

shop.linuxnewmedia.com

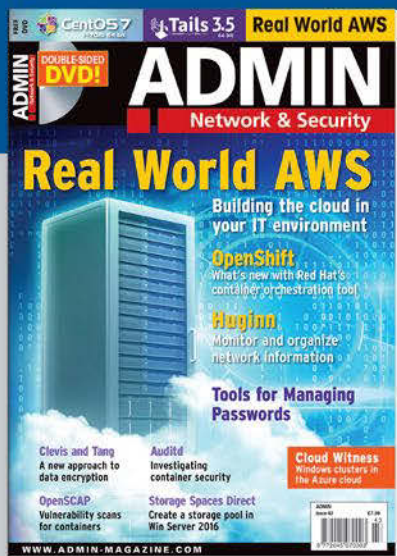
Want to subscribe?

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

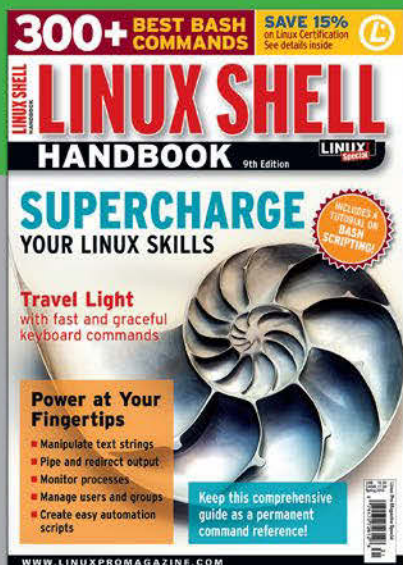
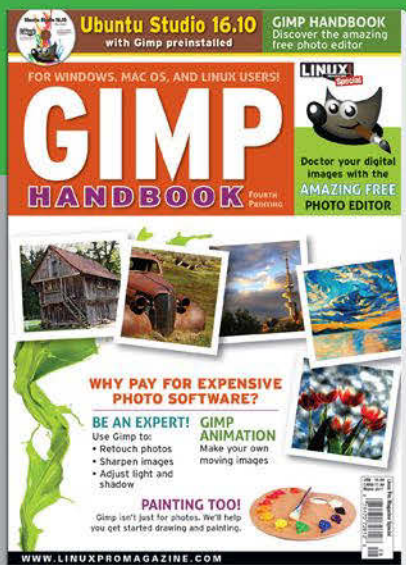
Discover the past and invest in a new year of IT solutions at Linux New Media's online store.

shop.linuxnewmedia.com

## DIGITAL & PRINT SUBSCRIPTIONS

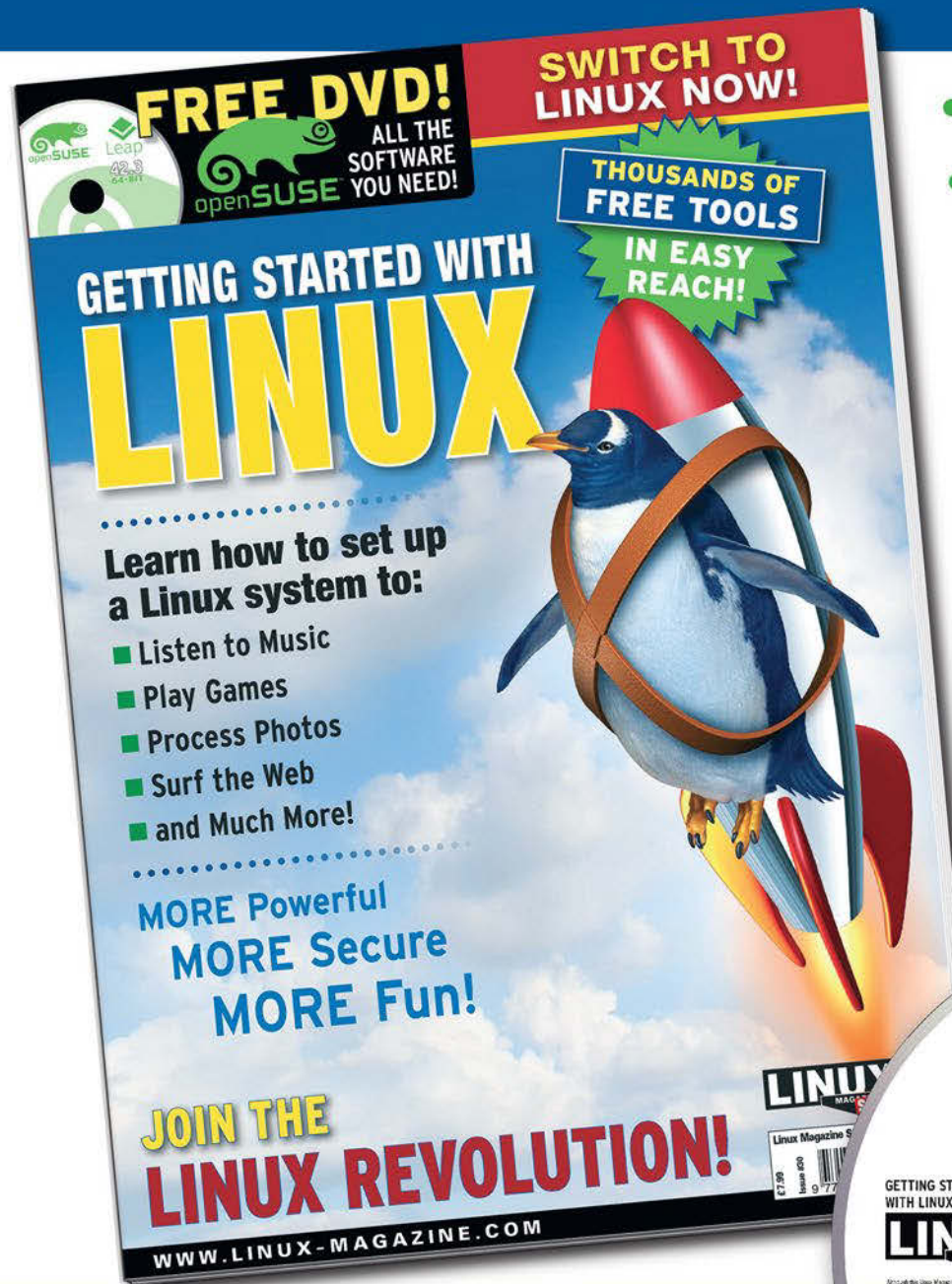


## SPECIAL EDITIONS



# Help your friends and colleagues make the switch to Linux!

This single issue shows beginners how to:



- Install Linux
- Download and install free software for your Linux system
- Create documents and spreadsheets
- Play games
- Surf the web
- Process photos
- Play music and videos
- and much more!



## HELP OTHERS JOIN THE LINUX REVOLUTION!

ORDER ONLINE:  
[shop.linuxnewmedia.com/specials](http://shop.linuxnewmedia.com/specials)

Many Linux users secretly hope Linux will one day be the only operating system in the world, but we all know that day hasn't happened yet. Talking to other systems is part of the challenge of working with Linux. This month, we introduce you to NitroShare, an easy tool that lets you exchange files and directories across Linux, Windows, Mac OS, and Android systems.

And speaking of Android...what is it? A whole different system, or just another kind of Linux? The beauty of open source software is that you can mold and customize it anyway you like, but despite the similarities, coding for the Android environment is not exactly like coding for Linux. This month we show you how to build an Android app with minimal programming using Jasonette, a tool that lets you create a complete Android application with a single JSON file.

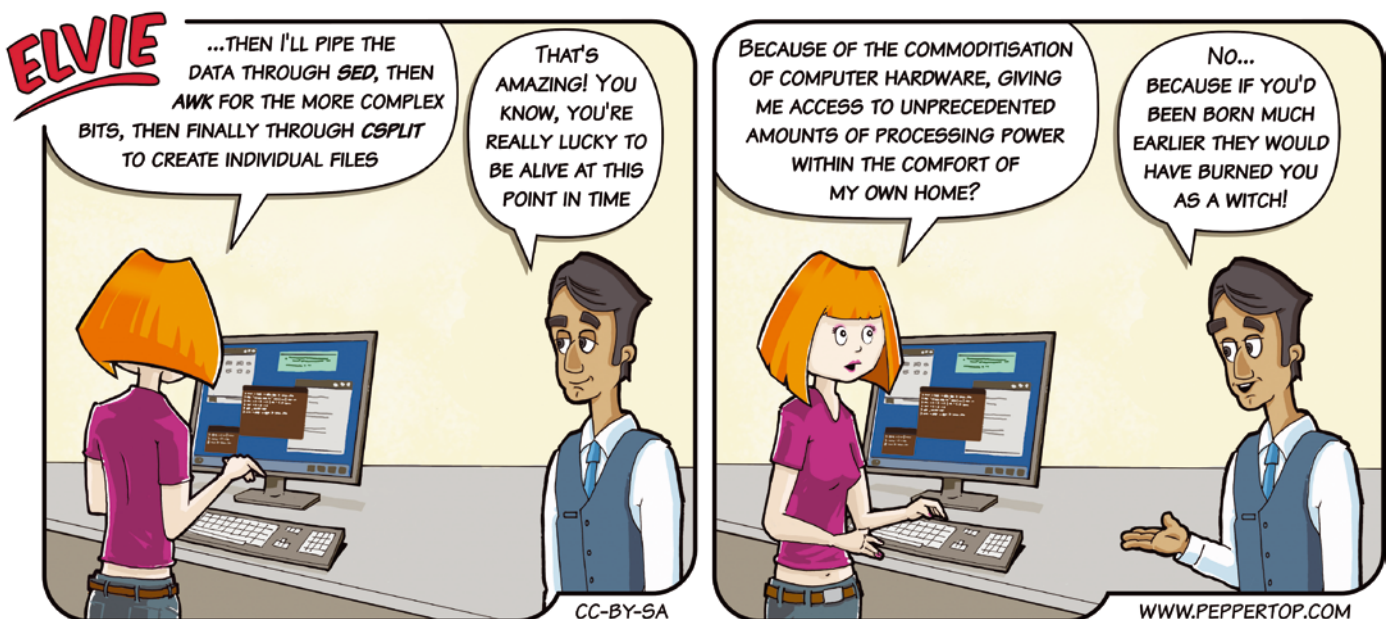
What about Android and Linux working together, like Master and Blaster? Have you ever wished you could use your phone's GPS capabilities to add location services to your computer? We've got that too. Read on to learn how to set up the Linux gpsd daemon to receive GPS data from an Android device.



Image © Olexandr Monoz, 123RF.com

# LINUXVOICE ▶

<b>Jasonette</b>	<b>70</b>
<i>Dmitri Popov</i>	
Build Android apps with a minimum of coding.	
<b>Doghhouse – Subutai</b>	<b>75</b>
<i>Jon “maddog” Hall</i>	
Be a catalyst and help others break the bonds of proprietary software.	
<b>NitroShare</b>	<b>76</b>
<i>Ferdinand Thommes</i>	
This handy app simplifies the exchange of files and directories between computers with Linux, Mac OS, Windows, and Android.	
<b>Nootka</b>	<b>80</b>
<i>Hartmut Noack</i>	
With Nootka and your guitar, you can learn to recognize musical notes in classical notation.	
<b>FOSSPicks</b>	<b>84</b>
<i>Graham Morrison</i>	
Graham checks out VLC 3.0, MenuLibre 2.1.5, Texttop, Flameshot, Chomper, Godot 3.0, and much more!	
<b>Tutorials – GPSD</b>	<b>90</b>
<i>Paul Brown</i>	
Even though your computer does not come with its own GPS, you can hack one onto it using a mobile phone – if you are ready for some fiddling.	



# Instant Android Apps

Jasonette makes it supremely easy to build simple and advanced Android apps with a minimum of coding.

BY DMITRI POPOV

For someone who doesn't write code for a living, creating even the simplest Android app can be a daunting proposition. However, for those of us with modest coding skills, there is Jasonette [1]. This ingenious solution makes it possible to create a full-featured Android app using a single JSON file that defines the app's basic structure and behavior. A generic Jasonette Android app pulls this JSON file and renders it as native Android components.

Creating a Jasonette-based app still requires some basic coding skills, but writing a JSON file is significantly easier than writing Java or Kotlin code. Also, building the Jasonette app in the Android Studio IDE requires only a few simple steps. Better still, because the JSON file controls the app's appearance and behavior, you can easily update and improve it by editing the JSON code instead of rebuilding the entire app.

### Preparatory Work

Before you start, you need to do some preparatory work. First, download and install Android Studio [2] on your machine. If you happen to use

openSUSE, installing Android Studio is a matter of running the following commands:

```
sudo mv android-studio-xx-xxx-xxx.zip /opt
cd /opt
sudo unzip android-studio-xx-xxx-xxx.zip
sudo chown -R root:root android-studio
sudo ln -s /opt/android-studio/bin/studio.sh \
/usr/local/bin/android-studio
```

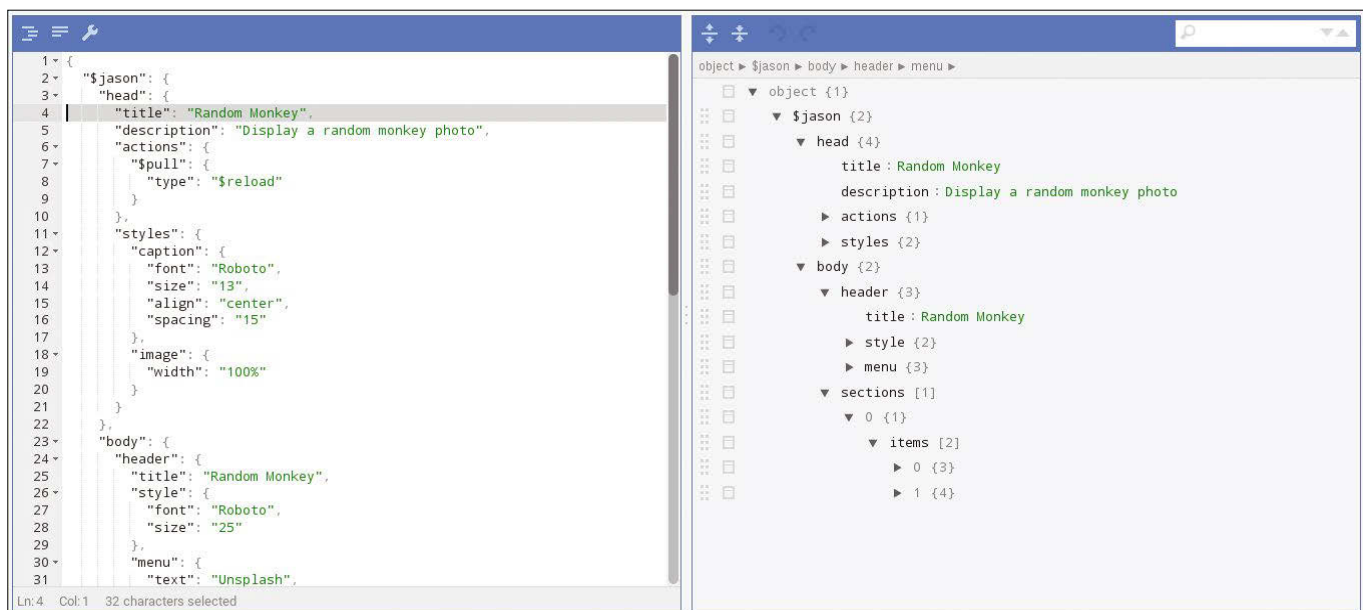
You don't need to use Android Studio to build an app just yet, though. To preview your JSON files directly on your Android device and test your app's basic functionality without building it in Android Studio first, you can use the Jason browser [3].

Instead of a regular text editor, you can use a dedicated tool for JSON code, such as JSON Editor Online [4], which runs directly in the browser, or the Elegant JSON Editor [5] Firefox add-on (Figure 1).

### Creating a JSON File

To get a basic understanding of what makes Jasonette tick and how JSON code works, take a look at the simple Jasonette app in Listing 1.

Figure 1: Elegant JSON Editor adds JSON editing capabilities to Firefox.



### Listing 1: Simple Jasonette App

```

{
  "$jason": {
    "head": {
      "title": "Random Monkey",
      "description": "Display a random monkey photo",
      "actions": {
        "$pull": {
          "type": "$reload"
        }
      },
      "styles": {
        "caption": {
          "font": "Roboto",
          "size": "13",
          "align": "center",
          "spacing": "15"
        },
        "image": {
          "width": "100%"
        }
      }
    },
    "body": {
      "header": {
        "title": "Random Monkey",
        "style": {
          "font": "Roboto",
          "size": "25"
        },
        "menu": {
          "text": "Unsplash",
          "href": {
            "url": "https://unsplash.com",
            "view": "web"
          },
          "style": {
            "font": "Roboto",
            "size": "19"
          }
        }
      },
      "sections": [
        {
          "items": [
            {
              "type": "image",
              "url": "https://source.unsplash.com/random/800x600?monkey",
              "class": "image"
            },
            {
              "type": "label",
              "text": "Here is a random monkey photo",
              "class": "caption"
            }
          ]
        }
      ]
    }
  }
}

```

This simple JSON code defines the basic structure and behavior of the app that fetches and displays a random photo of a monkey. Similar to an HTML page, the JSON file's `head` section (Listing 2) specifies general properties such as the app's title, description, and actions:

The `$pull` action in the header enables the pull-to-refresh action. The `styles` section is used to specify styling for various interface elements. In this example, it is `caption` for image text, and `image` for the image element. The `body` section (Listing 3) includes the three main elements of the app's interface: `header`, `menu`, and `sections`.

The `header` section in Listing 3 defines the app's header (the fixed bar at the top of the screen). In this case, the header contains a `title` and a `menu` item that opens the specified URL in the default browser. Note that in addition to the global styles, it's also possible to define inline styles for individual elements. For example, an inline style right after the `title` element defines the appearance of the header title. The `sections` part contains an array of items. In this example, the array consists of two items: The first one is the

### Listing 2: head Properties

```

"$jason": {
  "head": {
    "title": "Random Monkey",
    "description": "Display a random monkey photo",
    "actions": {
      "$pull": {
        "type": "$reload"
      }
    },
    "styles": {
      "caption": {
        "font": "Roboto",
        "size": "13",
        "align": "center",
        "spacing": "15"
      },
      "image": {
        "width": "100%"
      }
    }
  },
  ...
}

```

## Listing 3: body Properties

```

"body": {
  "header": {
    "title": "Random Monkey",
    "style": {
      "font": "Roboto Bold",
      "size": "25"
    },
    "menu": {
      "text": "Unsplash",
      "href": {
        "url": "https://unsplash.com",
        "view": "web"
      },
      "style": {
        "font": "Roboto",
        "size": "19"
      }
    }
  },
  "sections": [
    {
      "items": [
        {
          "type": "image",
          "url": "https://source.unsplash.com/random/800x600/?monkey",
          "class": "image"
        },
        {
          "type": "label",
          "text": "Here is a monkey photo",
          "class": "caption"
        }
      ]
    }
  ]
}

```

image, and the second one is its text label. And that's all there is to it.

To test your first app, you don't have to host the JSON file on your own server. Instead, you can use the [jasonbase](#) [6] JSON hosting service. Create a new file, give it a name, paste the code, and save the changes. Make note of the created JSON file's URL, launch the Jason app on your Android device, enter the JSON file's URL, and press Go. This adds a new entry, and you can launch the app by tapping on it (Figure 2).

### Building a Photo Portfolio App

With a few simple tweaks, you can transform the simple example app into a portfolio for displaying photos stored on your server. For this app, you need a web server for hosting a JSON file and photos. To keep things tidy, create a dedi-

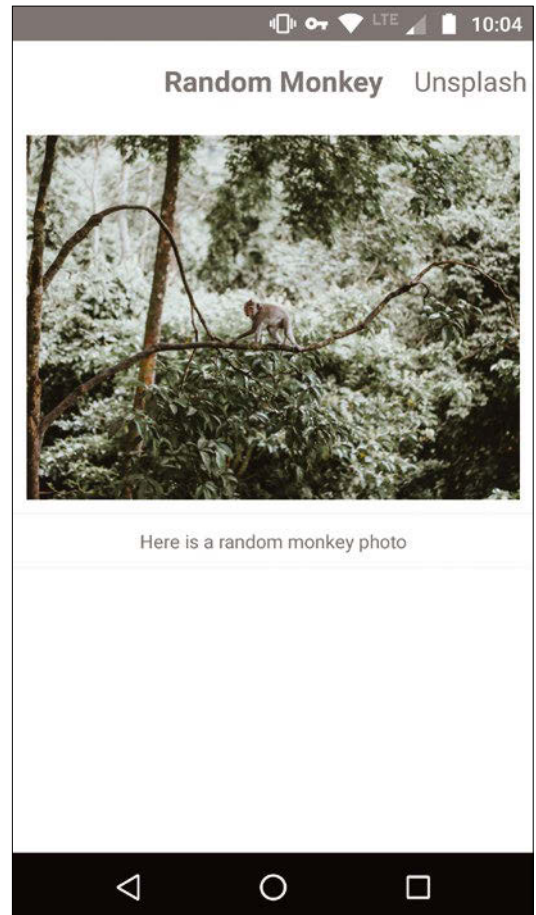


Figure 2: Random Monkey Android app in action.

cated directory on the server for storing photos and the JSON file (e.g., `photoapp`). Before you copy photos into that directory, you should resize and recompress them. This will make your app run faster and reduce the amount of data that needs to be transferred. The perfect tools for the job are `jpeg-recompress` [7] and ImageMagick's `mogrify`. With these two tools installed on your system, you can use the following command to recompress or resize all JPG files in the current directory:

```

for file in *.jpg; do jpeg-recompress Z
  --quality high $file $file; done
mogrify -resize "1600>" *.jpg

```

With all the pieces in place, you are ready to start working on the JSON file. At first glance, all you need to do is to change the appropriate properties and add image and label items for each photo. This quick-and-dirty solution has serious drawbacks, though. All URLs are hard-wired into the JSON file, so every time you add a new photo, you need to create another item in the JSON file manually.

One way to solve this problem is to replace the JSON file with a PHP script that generates JSON data on the fly. Listing the entire PHP



script is not very practical, so instead, use the command

```
git@gitlab.com:dmpop/
jasonette-tokyo-taxi.git
```

to grab the source code of the Tokyo Taxi Lights [8] app. Then open the `json.php` script that powers the app.

The script is not particularly difficult to parse. It starts by reading all files with the specified file extension (by default it's JPEG) into the `$files` array:

```
$files = array();
foreach (glob($ext) as $file) {
    $files[] = $file;
}
```

To pick a specified number of random photos, the script uses the `shuffle()` function:

```
shuffle($files);
$files = array_slice($files, -$number);
```

The script then checks whether each photo has an accompanying `.txt` file. If the text file is detected, its content is used as a caption; otherwise, the file name is used as the caption:

```
$txt = pathinfo($file, PATHINFO_FILENAME).".txt";
if (file_exists($txt)) {
    $caption = rtrim(file_get_contents($txt));
} else {
```

```
    $caption =
        rtrim(pathinfo($file, PATHINFO_FILENAME));
}
```

Finally, the obtained and assembled data is used to generate a complete JSON file.

If you have a working knowledge of PHP, you can modify the script to fit your particular needs. Otherwise, you can modify the user-defined values only. Copy the resulting PHP script to the web server and start building your first Android app in Android studio.

Use the command

```
git clone https://github.com/Jasonette/
JASONETTE-Android.git
```

to clone the **JASONETTE-Android** app's source code. Launch Android Studio by running the `android-studio` command in the terminal and import the **JASONETTE-Android** project. Next, open the `app/res/values/strings.xml` file, locate the `<string name="app_name">Jasonette` `</string>` line, and replace Jasonette with the desired name (Figure 3). Then, replace the default URL in

```
<string name="url">https://jasonette.github.io/
Jasonpedia/hello.json</string>
```

with the actual URL to your JSON file.

Now prepare an image file in the PNG format for use as an app icon. In Android Studio, right-click on the `app | res` folder and choose *New | Image*

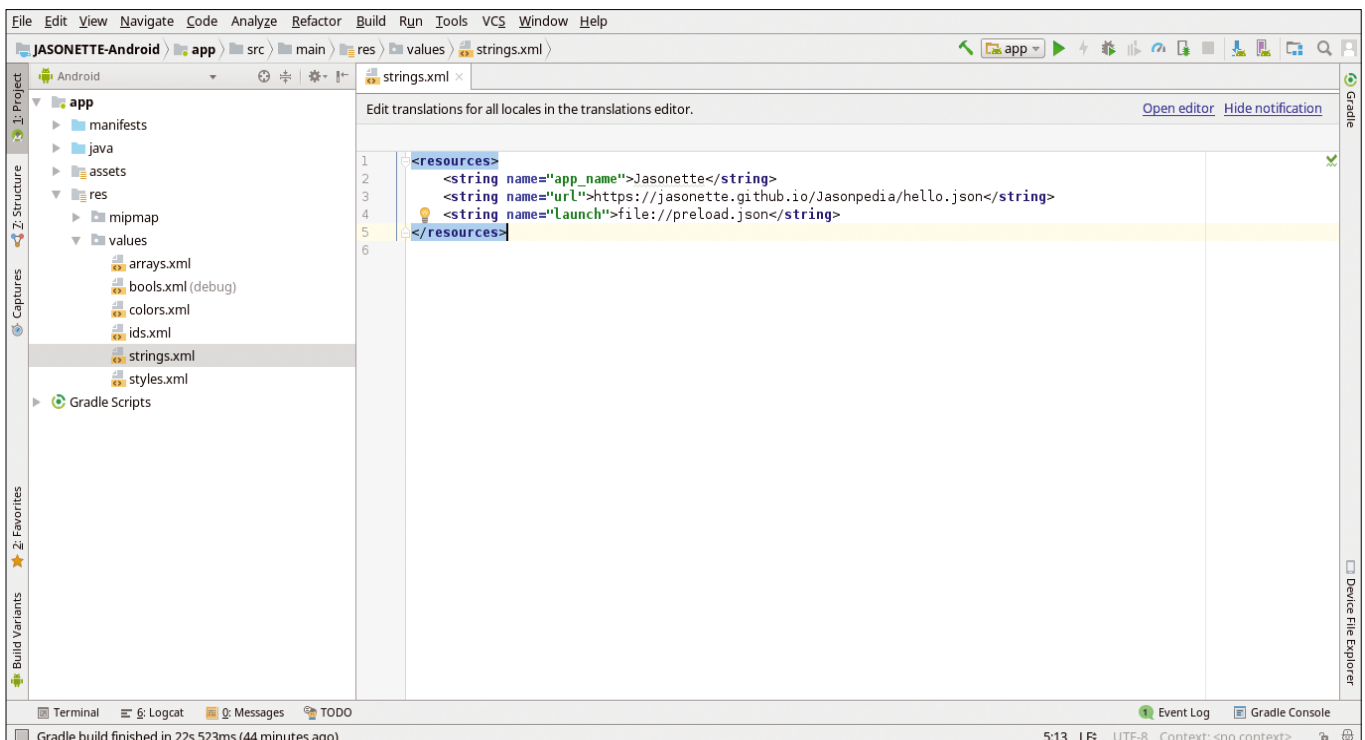


Figure 3: Editing default Jasonette values.

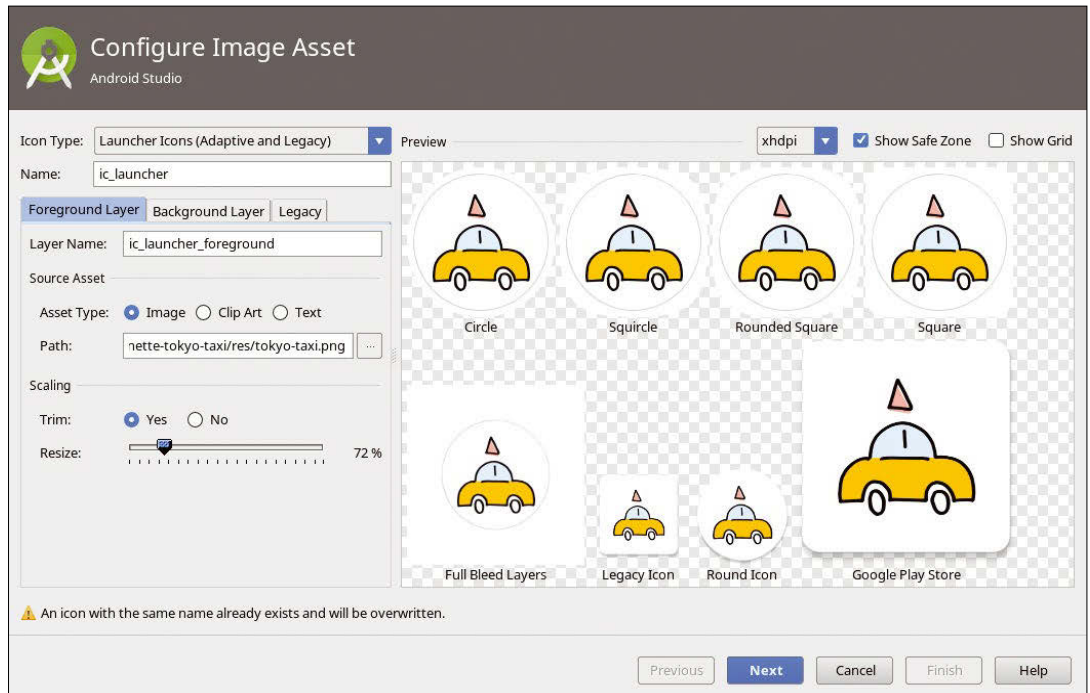


Figure 4: Creating an app icon.

Asset. Use the prepared PNG file to create and add an app icon (Figure 4).

Open the *Gradle Scripts | build.gradle(Module: app)* file, locate the `applicationId "com.jasonette."` line, and replace `com.jasonette` with the desired app ID. Press the *Sync Now* link to synchro-

nize the changes. Now, choose *Build | Build APK(s)* to build the app.

To install the resulting `app-debug.apk` package on an Android device (Figure 5), switch to *Settings | Security* and enable the *Unknown Sources* option.



Figure 5: Tokyo Taxi Lights app.

### Conclusion

Using Jasonette, you can build your first simple Android app in less than an hour, but despite its deceptive simplicity, Jasonette packs advanced features that make it possible to develop and build sophisticated apps. Make sure to check the Jasonette documentation [9] to get the most out of this excellent tool. ■■■

### Info

- [1] Jasonette: [jasonette.com](http://jasonette.com)
- [2] Android Studio: [developer.android.com/studio/index.html](http://developer.android.com/studio/index.html)
- [3] Jason: [play.google.com/store/apps/details?id=com.jasonette.jason](http://play.google.com/store/apps/details?id=com.jasonette.jason)
- [4] JSON online editor: [jsoneditoronline.org](http://jsoneditoronline.org)
- [5] Elegant JSON Editor: [addons.mozilla.org/en-US/firefox/addon/elegant-json-editor](http://addons.mozilla.org/en-US/firefox/addon/elegant-json-editor)
- [6] jasonbase: [www.jasonbase.com](http://www.jasonbase.com)
- [7] jpeg-recompress: [github.com/danielgtaylor/jpeg-archive](https://github.com/danielgtaylor/jpeg-archive)
- [8] Tokyo Taxi Lights: [gitlab.com/dmpop/jasonette-tokyo-taxi](https://gitlab.com/dmpop/jasonette-tokyo-taxi)
- [9] Jasonette documentation: [docs.jasonette.com](http://docs.jasonette.com)

# MADDOG'S DOGHOUSE

Be a catalyst and help others break the bonds of proprietary software. BY JON "MADDOG" HALL

## Open source catalyst

Over the past 50 years, I have worked for many companies and was a consultant for many government agencies and nonprofits. Since July 2017, I have been the CEO of OptDyn, the creators of Subutai open source peer-to-peer cloud software. I wrote about Subutai in the February 2018 column and thought I would add an update.

While the cost of creating a business or even launching a new product for an existing business has gone down over time, it is still amazing how much it costs and how long it takes.

In chemistry, there is a term "catalyst," a substance that speeds up a reaction often without actually taking part in the reaction itself. Platinum is a catalyst that is often used in fuel cells to allow the hydrogen and oxygen to fuse together, releasing electricity with a byproduct of water.

Open source is another catalyst, allowing companies to start up without needing huge investments in cash to buy closed source development tools, library licenses, databases, and other components to the overall solution.

Open data and open culture are both additional catalysts for business, lowering the costs and speeding up the ability of fledgling companies and products to come to market.

You have to be careful, however, of the difference between "free" as in beer and "free" as in freedom. Often the thing that starts out "low cost" limits what you can do later or has hidden costs that become visible as the use increases.

In physics, Newton's laws of motion state that a body at rest tends to stay at rest and a body in motion tends to stay in motion unless they are acted on by some outside force. The shorthand name for this is "inertia."

We have inertia in the software world also. The law seems to be written: "People who use some particular software will continue to use that particular software unless some outside force makes them use other software." People use a particular piece of software because they are familiar with it and how it works, and even if there is another piece of software that is demonstrably better, they still stick with "the software they know."

What does it take to make them break the mold and go to another piece of software? Perhaps the software company that made the software goes out of business or the person has to

pay the price of an upgrade (forced by an incompatibility with a new version of the operating system).

Of course there is also the social effect – all of the person's friends use one make of software and that person wants to fit in, to be able to show their friends what they learned or ask questions about problems in hope of getting an answer. Or they want to use software that is very expensive, to fit into a certain crowd.

If you are reading this article, then you are probably someone who is outside of the realm of inertia and who has found your catalyst a long time ago in free software.

Despite the use of the "cloud," there are still a *LOT* of local machines used, or should we say "not used." Many machines that are in our homes, offices, and schools are not used as efficiently as they might be, and therefore we use more commercial cloud resources than we might otherwise use.

This is a good time to investigate the use of Subutai open source peer-to-peer cloud software. Set it up on your own machine and help others in your community (whether it be at school, work, or home) to use the resources you already have for CPU, memory, and storage. It does not cost you anything and could save you a lot of money.

You can download it from [subutai.io](http://subutai.io). If you wish to buy or sell resources you need or have extra, you can do that at one of the Subutai Bazaars on the net.

Subutai uses a lightweight digital currency called GoodWill to buy and sell resources. You can also earn GoodWill by doing a number of other activities, such as writing Subutai Blueprints.

Blueprints are files that tell Subutai how to run programs, and allow people to run these programs by clicking on the Blueprint name just as you might click on any other application that you have on your system. Blueprints are not very hard or time-consuming to create. A Blueprint for Mattermost (an open source clone of Slack) took only three hours to create.

There may be a good business in helping people use their own machines more efficiently, or helping them move their application to a cloud model that allows them to use resources from anywhere. Go to the Subutai website and learn how to write a Blueprint. ■■■



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®.

# In All Directions

This cool tool helps you exchange files and directories between computers with Linux, Mac OS, Windows, and Android. **BY FERDINAND THOMMES**

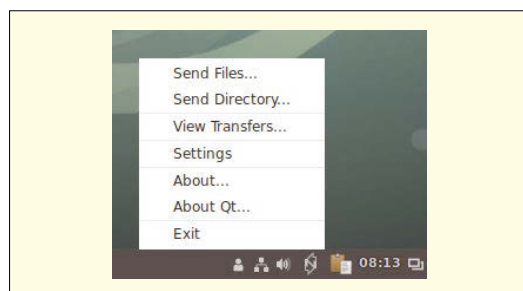
**T**here are many tools for exchanging data on heterogeneous local networks. NitroShare promises to make this task as easy as possible. In addition to Linux, NitroShare also supports Mac OS, Windows, and Android.

In this article, my test included a network of computers running Debian “Sid,” Windows 10, Ubuntu, Ubuntu MATE, Fedora, and the Android app. Users with Debian Sid, Ubuntu, Fedora, Gentoo, and openSUSE will find the software in their respective repositories; for Arch Linux users, check out the AUR user archive. If necessary, compile the program for other distributions from the source code.

## Qt5 as a Basis

Packages for Nautilus (Gnome), Nemo (Cinnamon), and Caja (MATE) are available for integration in file managers. Although NitroShare is based on the Qt5 framework, not all distributions have appropriate extensions. Packages for integration into Dolphin and Konqueror were only available for openSUSE as *dolphin-plugin-nitroshare* and for Fedora as *nitroshare-kservice*. Fedora also offers a plugin for use in the terminal.

The installation on Debian and Fedora is quickly done (Listing 1); the package itself is only 150KB with just one dependency on a small 40KB Qt web server. If you do not want to use the Personal Package Archive (PPA) in Ubuntu, use the same command as in Debian. For Windows, executables for 32- and 64-bit systems are available on



**Figure 1:** NitroShare is located in the toolbar’s system area under Debian.

the project’s website [1], and a DMG package is available for Mac OS from version 10.8.

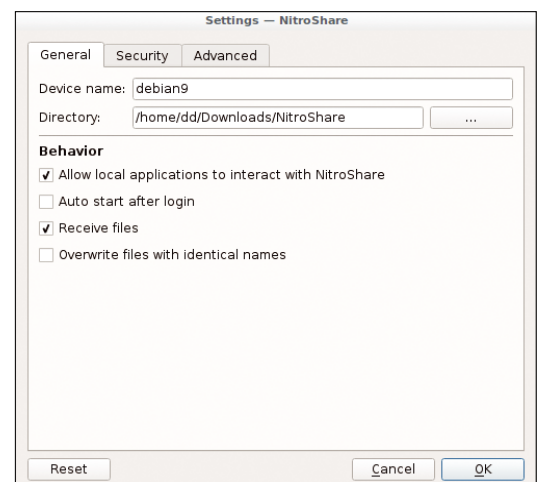
## Immediately Ready for Operation

After installation and an initial start from the respective menu, the program is installed in the system tray (Figure 1). From there, it offers the exchange of files and directories with computers on the LAN on which the software is also installed. NitroShare automatically detects the corresponding devices and offers them as transfer targets.

The software is ready for use immediately after installation; no further configuration is required. In the Settings menu, you have the option of configuring a number of settings depending on your distribution and desktop (Figure 2 and Figure 3).

Checking *Auto start after login* causes the program to launch whenever the system is booted. Automatic detection of other devices can take up to five minutes the first time. At the end of this phase, the tool displays a message with the results.

In the Settings dialog, you can also specify a name other than the hostname for each device and a default folder for downloads in the *General* tab. Under *Advanced*, you can change the port, buffer, time out, and interval, if necessary.



**Figure 2:** NitroShare offers more options in Settings under Debian than anywhere else.

**Listing 1: NitroShare Installation on Debian and Fedora**

```
# Debian:
apt update
apt install nitroshare

# Fedora
sudo dnf install nitroshare

# Ubuntu-PPA:
sudo apt-add-repository ppa:george-edison55/nitroshare
sudo apt-get update
sudo apt-get install nitroshare
sudo apt install nitroshare-nautilus

# Arch from the AUR:
wget https://aur.archlinux.org/cgit/aur.git/snapshot/nitroshare.tar.gz
tar xf nitroshare.tar.gz
cd nitroshare
makepkg -sri
```

**Problems**

In my lab, I looked at Linux-Linux and Linux-Windows connections, and vice versa, as well as Android-Linux and Android-Windows connections. NitroShare failed to cooperate with a workstation on Debian Sid, with failures in both directions and an error message regarding SSL that did not make sense. The problem resulted in a bug report on GitHub [3].

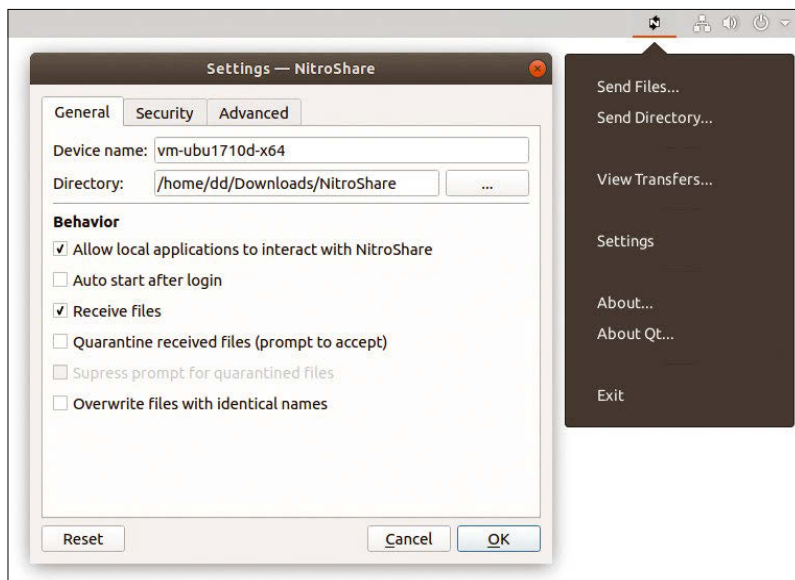
**Encryption Is Possible**

Although not normally necessary on your own network, the software also offers the possibility to encrypt transfers via Transport Layer Security (TLS) in the *Security* tab. A description of the corresponding configuration (Figure 4) can be found on the NitroShare wiki [2].

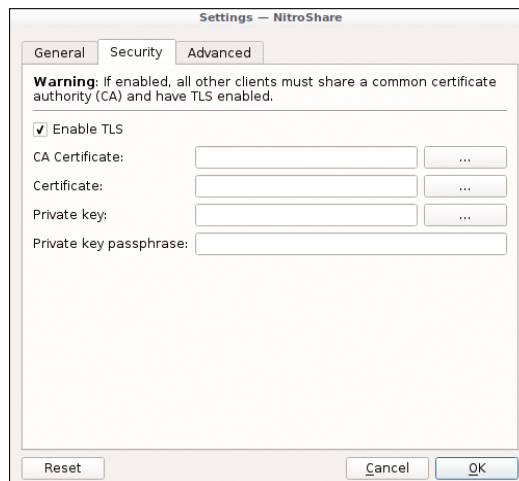
The menu also offers the *Send Files* and *Send Directory* options, which allow you to select files or directories and then specify the desired transfer destination. During data transfer, the software displays its history. You can close this dialog without affecting the transfer. After the transfer has completed, you see a message to this effect.

NitroShare does not limit the size of the data sent and can thus theoretically achieve high speeds on networks designed for this purpose. Behind the scenes, the program uses IPv4 broadcasts to send User Datagram Protocol (UDP) packets to other devices on the LAN. These contain information that allows other devices to initiate transmissions without having to worry about broadcast addresses and the like. By default, NitroShare sends all UDP packets on port 40816 to any active network interface with an IPv4 broadcast address. You can change the port individually in the settings, if necessary.

The application worked perfectly in all combinations and in all directions on Debian Sid,



**Figure 3: Ubuntu MATE offers some options that aren't present in Debian.**



**Figure 4: NitroShare offers the possibility of encrypting data in transit using TLS.**

Ubuntu, and Windows notebooks and on an Android 8.1 smartphone. Controlling the application directly with its own tools could not have been easier.

Integration with the Nautilus file manager on Ubuntu (Figure 5) and Caja on Ubuntu MATE did not work: The context menus did not show any corresponding entries.

I measured the speed of NitroShare with a 1.5GB file, which I sent wirelessly with Debian Sid and Ubuntu three times in both directions between the two notebooks. It reached transmission rates between 90 and 110KBps. With many small files totaling around 700MB, the data rate remained roughly the same.

### Conclusions

Work on NitroShare has been in progress for six years now, so it would appear that any teething problems have now been eliminated. However, this was not always the case in the test, although

not being able to access the Debian Sid workstation might not be the fault of the program itself. Nevertheless, it would be practical to copy the (in this case long) error message instead of typing it. The reason the integration in Nautilus and Caja did not work was still unclear after completing the test.

Despite these few problems, NitroShare does what it promises: After the install, you can get started immediately without having to configure any settings. Transfers take place quickly.

Android devices are limited to the sender role; the software does not offer mobile devices as targets on other clients. To do this, some Android apps offer the option of sending files directly via NitroShare. During the test, I found corresponding entries in Google Docs, Gallery, and Photos, but there may be more of them. Nevertheless, as far as Android is concerned, KDE Connect – which is even available on Gtk desktops – seems to be the more mature solution. ■■■

### Info

- [1] NitroShare: <https://nitroshare.net>
- [2] TLS: <https://github.com/nitroshare/nitroshare-desktop/wiki/TLS>
- [3] Bug report: <https://github.com/nitroshare/nitroshare-desktop/issues/155>

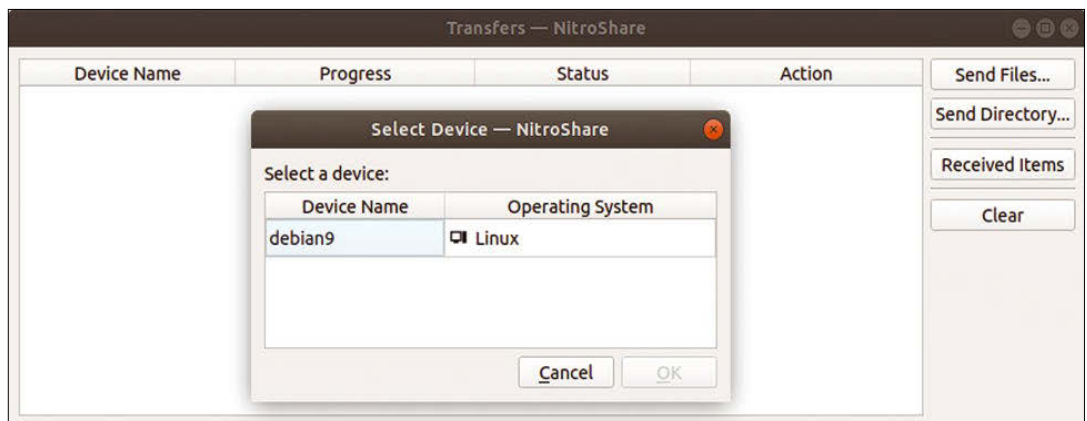
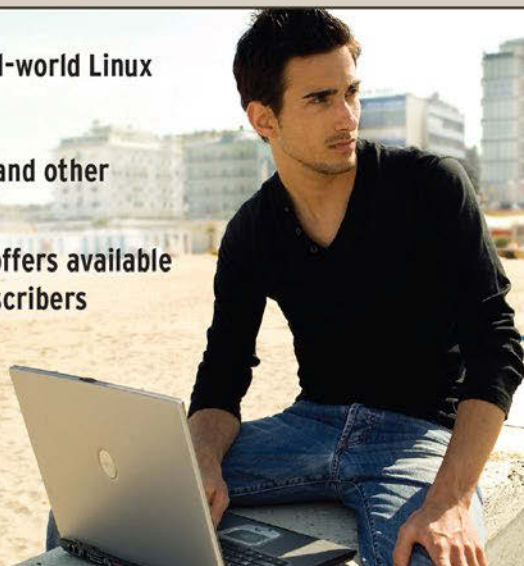


Figure 5: Ubuntu offers a PPA with up-to-date versions of NitroShare. However, embedding in Nautilus does not work.

# LINUX UPDATE

Need more Linux? Our free Linux Update newsletter delivers insightful articles and tech tips to your mailbox twice a month.

- Original articles on real-world Linux
- Linux news
- Tips on Bash scripting and other advanced techniques
- Discounts and special offers available only to newsletter subscribers



# COMPLETE YOUR LIBRARY

Order a digital archive bundle and save at least **50% off** the digisub rate!



**ORDER YOURS TODAY!**  
[shop.linuxnewmedia.com](http://shop.linuxnewmedia.com)



You get an **entire year of your favorite magazines** in PDF format that you can access at any time from any device!

# String Theory – Fast guitar training for classical music

With Nootka and your guitar, you can learn to recognize musical notes in classical notation and train for precision and speed.

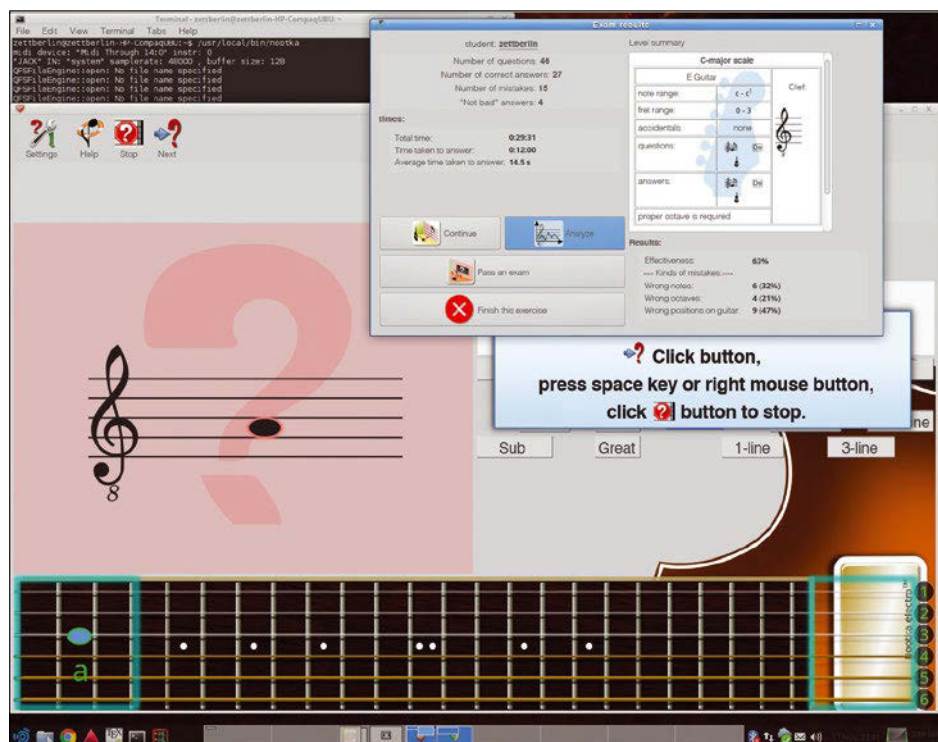
BY HARTMUT NOACK

**L**inux has many small tools that allow music enthusiasts to practice reading musical notation. Nootka [1], by developer Tomasz Bojczuk, offers an extensive program that helps guitarists learn notation (Figure 1).

The program, which runs on many platforms (see the “Installation” box), trains basic musical knowledge and the ear. The software can only display and play whole notes and is thus not suitable as a tool for composing music. Nootka does not allow for odd metrics, rhythms, and subtleties such as triplets, nor does it support chords. Instead, it helps you concentrate fully on learning the difficult art of recognizing notes in classical notation.

The application is integrated into the JACK professional sound system, which makes it easy to connect musical instruments to the program. Some tasks require you to play or sing notes displayed in

**Figure 1:** With a variety of exercises, playful interactions, and motivating evaluations, Nootka helps guitarists learn classical notation.



musical notation, which is something JACK does easily. If you select singing mode, it is also possible to practice with a keyboard, for example, by connecting a software synthesizer like Yoshimi or Calf to the Nootka JACK port.

## Classical Notes

Nootka covers the whole range of classical European music, as well as rock and pop, although the restriction to simple keys in major and minor might become boring in the long run. Additionally, the software lets you analyze and practice ecclesiastical modes, 12-tone compositions, and other special scales (i.e., scales containing notes whose combined use in chords and melodies creates a matching harmonic effect) by freely entering the notes instead of the defaults.

Localization of the program shows a love of detail: You can adjust the conventions for naming

notes to various international languages, which the program then takes into account (Figure 2), and translates the user interface to an acceptable quality, although grammatical errors sometimes slip through.

The program teaches notation for all notes used in equal temperament [2] and pure intonation [3] tuning, although Nootka is mainly geared to standard guitar tuning. If you use a software synthesizer such as Yoshimi, you can use pure intonation, with the correct distinction between tones such as E-sharp and F, and learn and practice the correct notation for such tunings. In single-note mode, you can display enharmonic equivalents [4], wherein some notes have different names on different scales, although they sound the same in temperament tuning (as is customary with piano and guitar). Thus C-sharp on the



## Installation

Nootka relies on Qt5 and is available on many platforms, including Linux, Windows, BSD, and Mac OS. Also, you can install a Nootka version for Android via an APK, which you can find on the website and in the Google Play store. The package managers of common Linux distributions offer the software, and the current stable version is available as a DEB, RPM, and archive for Arch Linux.

If you don't want to use the package manager, you can download an AppImage of the current stable version from the website. Although it started under Ubuntu Studio 16.04 without any problems, it didn't run as stably as the other packages. Crashes occurred when changing settings for the user interface and audio port.

To build a current version from source, you not only need build tools like Make and GCC, but also the latest development packages for Qt5 and Cmake. Under Ubuntu 16.04, the three-phrase `cmake, make, and make install` ran without problems for the current development version used in the test (1.4.4.4).

guitar is identical to B, but can actually be played with a different frequency on a violin.

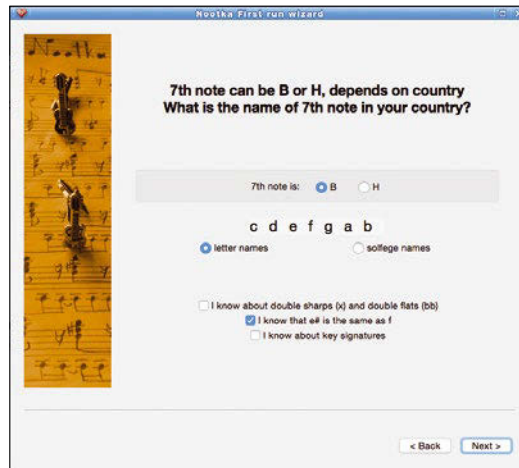
The supposedly simple concept of Nootka thus turns out to be a simple training system for solid basic knowledge, which can be used for very demanding compositions (Figure 3).

### Step by Step

The tasks within the scope of training prove to be quite demanding: Even for the simplest level, which is all about empty guitar strings, you need to know the position of the six notes on the staff. The staff shows no preview of the note values in exercise and exam modes. If a question asks you to put a note on the fretboard at the bottom, a preview would be useful for beginners

If you don't have any experience with notation, you should first use a function that is not necessarily obvious in the basic view of the program: Nootka displays detailed information about the notes played on a connected instrument, which allows beginners to explore the connections between notes played on a guitar and notation in free mode. The program recognizes each tone with remarkable accuracy and displays it as a note.

The tutorials provided are all based on the assumption that you know the position of the notes in sheet music; it is all about training precision and speed, rather than learning the basics. Depending



**Figure 2:** In the basic settings, you can decide whether Nootka should use AHC notation (e.g., as in Germany) or the ABC standard. The Italian DoReMi nomenclature also can be configured.

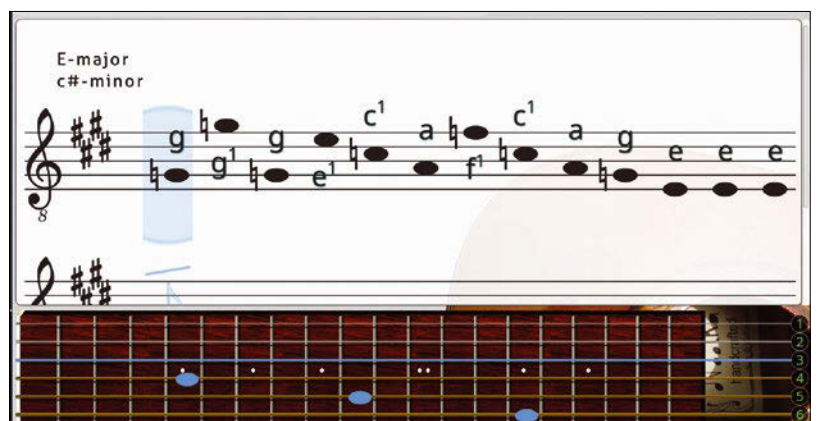
on your previous knowledge, you can reconfigure the tutorials as required (see the box titled "A Question of Settings").

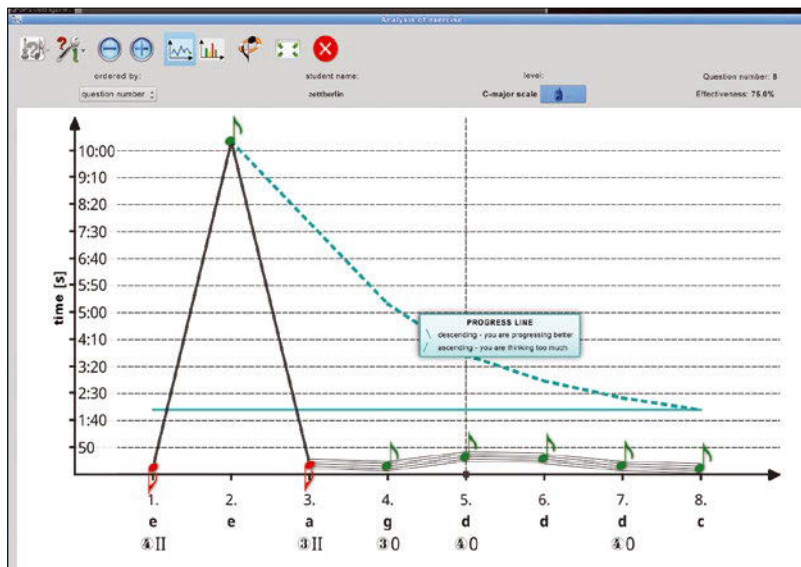
Although easy levels are available from the beginning to test your knowledge, beginners should first choose *exercise* rather than *exam*. Such tutorials do not set a time limit and simply ask one question after another on one subject.

You can disable the somewhat inconvenient way of asking the next question and confirming the answer in the toolbar by pressing the Spacebar and Enter keys instead. After a short delay, the program automatically asks new questions and accepts the inserted or played note as the answer. You can quit an exercise at any time by pressing the *Stop* button in the upper left corner. Clicking on *Analyze* shows an overview of the quality of answers given and the time required (Figure 4).

During exercises, an automatic function proposes to test the same topic with tougher specifications when the answers reach a certain quality (Figure 5). If you answer most of the questions relatively quickly during testing, you will be shown a cute little certificate.

**Figure 3:** Nootka can display simple notation with letters along with standard notation, if desired.





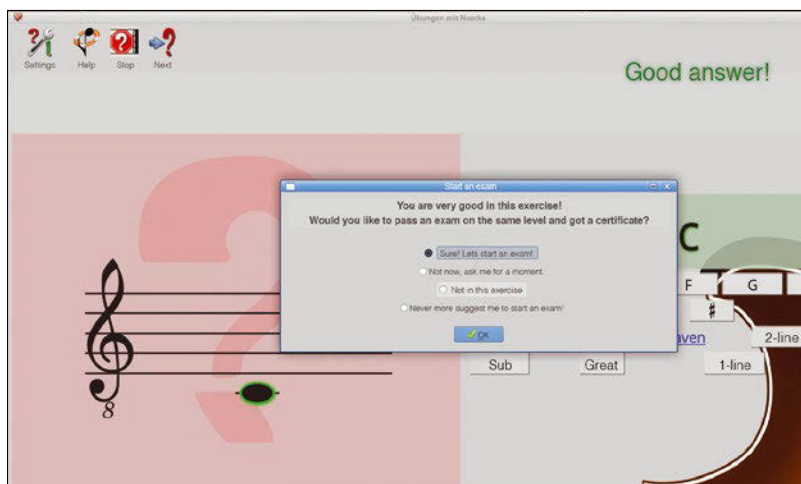
**Figure 4:** The analysis shows hints about the meaning of the curves when you mouse over an element.

### Alternatives

If Nootka doesn't quite meet your requirements, other projects offer additional possibilities. For example, the Java program ScoreDate [5] offers considerably less customization, but also allows rhythm training, including drum notes. Additionally, it does not limit its musical notation to whole notes. The last update of the program was in 2012, but the package from Ubuntu "Universe" works fine.

Impro-Visor [6], which is not necessarily suitable for beginners, is aimed primarily at ambitious jazz musicians, offering a wide range of practice possibilities. The program allows you to compile virtual backing bands for your own improvisations and comes with more than 400MB of musical material on board. Music examples by well-known jazz legends such as Dizzy Gillespie and Charlie Parker let you analyze the stylistic compositional methods of the genre. Contrary to what might be expected from the outdated website, the developers regularly maintain Impro-Visor. The current version is from June 2017.

**Figure 5:** When your performance reaches a certain level in exercises, Nootka offers to administer an exam.



### A Question of Settings

The overall appearance and behavior of Nootka largely depends on the settings for the selected level and those in *Appearance*. For example, in a level that is set to *single note*, the software only displays one note in the staff, with the exact name in the box to the right. Levels set to *Melody* present long lines with multiple notes, all of which are whole notes, but can assume pitches available in classical notation.

You can hide many elements of the interface to make room for others that expand accordingly. The sheet music and – in single note mode – the table of notes remain visible. In the level editor, you can select or complete training tasks of different difficulty levels and thematic focus. In this way, you can decide whether a tutorial should contain exercises for single notes only or for melodies. The software explicitly points out contradictory selections and does not offer an automatic mechanism to resolve conflicts, which means you have to resolve the problems yourself.

### Conclusions

All in all, Nootka appears logical and easy to use after familiarization with its extensive dialogs. The simple, clear design and the friendly responses offer motivation for long exercises. The free mode also lets you create notations for your own melodies. ■■■

### The Author

Hartmut Noack works as a lecturer, author, and musician in Celle and Hanover, Germany. He finds that free software and homemade music go great together. You can find some Creative Commons-licensed results of his work with free music software at <http://lapoc.de>.

### Info

- [1] Nootka: <http://nootka.sourceforge.net>
- [2] Equal temperament tuning: [https://en.wikipedia.org/wiki/Equal\\_temperament](https://en.wikipedia.org/wiki/Equal_temperament)
- [3] Pure intonation tuning: [https://en.wikipedia.org/wiki/Just\\_intonation](https://en.wikipedia.org/wiki/Just_intonation)
- [4] Enharmonic equivalents: <https://en.wikipedia.org/wiki/Enharmonic>
- [5] ScoreDate: <http://scoredate.sourceforge.net>
- [6] Impro-Visor: <https://www.cs.hmc.edu/~keller/jazz/improvisor/>

THE COMPLETE

# LINUXVOICE

ARCHIVE DVD: ALL 32 ISSUES!

LINUXVOICE

ARCHIVE DVD: ALL 32 ISSUES!

ISSUE 200 JULY 2017

LINUX  
MAGAZINE

ARCHIVE DVD: ALL 32 ISSUES!

DVD  
ROM

3,400  
PAGES OF  
LINUX AND OPEN SOURCE LOVE!

3,400  
PAGES OF  
LINUX AND OPEN SOURCE LOVE!

LINUX AND OPEN SOURCE LOVE!

Since April 2014, Linux Voice has showcased the very best that Free Software has to offer. Now you can get it all on one searchable DVD.

Includes all 32 issues in E PUB, PDF, and HTML format!

Order now!

[shop.linuxnewmedia.com](http://shop.linuxnewmedia.com)



# 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**

## Media player

# VLC 3.0

There can't be many Linux users who haven't heard of or installed the venerable VLC. For many years, it's been an essential installation requirement, even if you don't play back movies on your Linux machine. It's just as good at playing YouTube videos and for listening to Internet radio, which is why the release of version 3 is such a milestone; it's perhaps one of the biggest updates to one of the most popular open source projects out there. This is reflected in the download

links. These prominently list iOS, Android, Mac OS, Windows, and even Windows Phone, while Linux users are challenged by asking their favorite packager rather than providing a direct download. There's not a snap or flatpak in sight, reflecting the origin of the majority of VLC users. Fortunately, for most distributions, the latest release can indeed be installed with a simple `sudo snap install vlc` without even asking your packager.

It's surprising that something called the "VideoLAN client" has

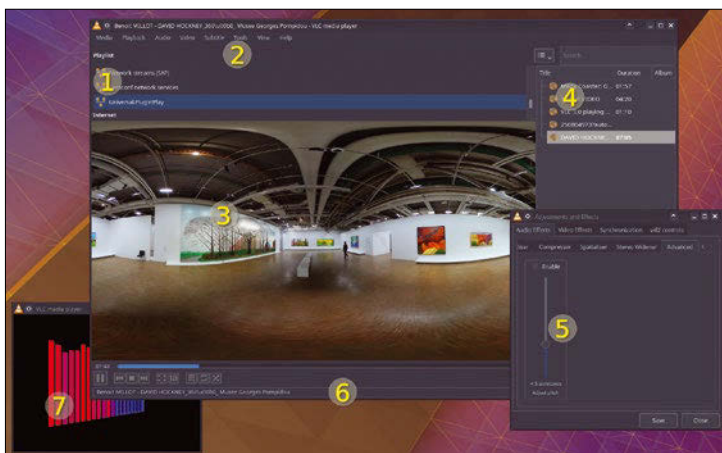
taken so long to get its best new feature: the ability to natively browse network filesystems such as Samba, SFTP, NFS, and FTP. You've always been able to paste an accessible URL into the network file requester, but you needed your desktop or background operating system to have mounted a remote filesystem first or delivered the data via its own transport mechanisms. As most of us like to store our media on remote devices, such as a NAS, this was always a problem. Now VLC can access these files without any extra help, and it makes a big difference, especially on Android.

While the VLC user interface looks largely the same as version 2 – complete with an even larger set of advanced settings – it can play back far more media. In particular, thanks to a considerable number of OpenGL acceleration additions on Linux, you can now play 4K and 8K videos at 60 frames per second, as well as 360-degree videos, such as those recorded for virtual reality headsets. Video playback

now includes support for the latest home-theatre buzzwords, including the ability to play back high-dynamic range video and ambisonics 3D audio, if you have the hardware configuration to make sense of these.

This is where another great addition makes more sense. You can now send video directly from your desktop VLC to a Chromecast plugged into your television. This is a fantastic upgrade simply because Linux (and open source!) lacked an easy and reliable way of doing this. It's a great way of storing and managing your movie collection on the go. And if you do plug your Linux box directly into the TV or amplifier, you can now choose to passthrough high-definition audio codecs, such as E-AC3, TrueHD, and DTS-HD. Because this is a version that's going to be around for some time (it's an LTS release), you can build your system around Linux and VLC and know it will remain working through various updates.

**Project Website**  
<https://www.videolan.org>



**1 Browse networks:** You can now access remote locations directly within VLC. **2 Chromecast support:** You can't see it, but these menus hide the option to beam your video to your TV. **3 360-degree video:** Play fully immersive video without resorting to Chrome. **4 Playlist improvements:** Stated simply, they now work, unlike with version 2.5. **5 Adjust pitch:** This new real-time audio effect is perfect for learning Prince's guitar solos. **6 HDR, 4K, and 8K playback:** Thanks to lots of new OpenGL acceleration, Linux can now be part of the extreme bandwidth and storage party. **7 Cross platform:** The same VLC will run on your Android device, your Linux box, and even any other computers you may have.

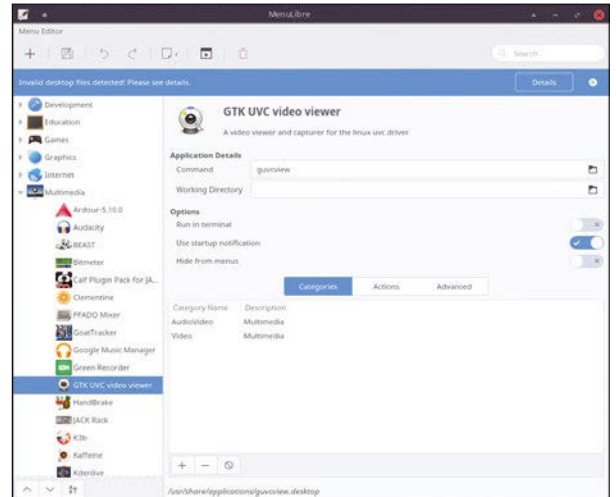
## Menu editor

# MenuLibre 2.1.5

Even though many of us don't use a launch menu for every application launch, they're still useful to see what's installed and for accessing forgotten things. This means that their associated menu editors aren't the most exciting part of the desktop. In fact, there's a good chance you've either never loaded one up to play with or haven't done so for a very long time. But they do perform an important function, especially when you're at the mercy of your chosen distribution's menu hierarchy, or worse, the whims of applications like Wine that seem to hook them into every sub-folder. A menu editor needs to be quick, powerful enough to accommodate your launch requirements, and most importantly, Desktop Entry Specification compliant. The

latter is important because it means you can use the same tool to edit your launch menu regardless of desktop environment or distribution.

This is what MenuLibre does. It has been tested with Budgie, Gnome, KDE (Plasma), LXDE, LXQt, MATE, Pantheon, Unity, and Xfce, and it lets you edit almost anything about the applications you find in the launch menu equivalents for those various desktops. You can change the icon for an application, for example, or set whether it runs within a terminal or is hidden from menus. You can also set the command to launch the application itself and test the validity of the desktop file generated by MenuLibre when you save your changes. General menu management is also possible, with buttons to move



Get the same menu editor regardless of which desktop environment you use.

menu headings up or down through the hierarchy or sort alphabetically. It's a great tool that does everything you need, on almost any desktop, which is a great advantage if you like to make menus your own.

**Project Website**

<https://bluesabre.org/2018/02/06/menuibre-2-1-5-released/>

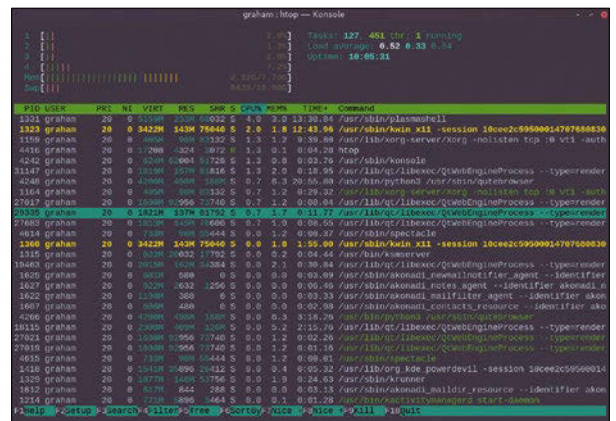
## Process monitor

# htop 2.1.0

There are few tools more effective than the venerable `top` command for process monitoring and control. This is because it can be run from almost anywhere that has a Linux command line and gives you an immediate and real-time overview of what processes are running; how much memory, swap, and CPU they're consuming; and the ability to kill a process by simply selecting it and issuing a command. Except that `top` is perhaps a little too venerable at times, with its minimalist and austere user interface written for 1984-era Unix rather than the cloud droplets of the 21st century. `htop` is a drop-in replacement for this fabulous command. While it adds lots of features lacking in the original, its

most critical asset is that it's still being updated, as this release shows. Version 2.1.0 is a great upgrade on what was already an excellent 2.0 release, adding process monitoring geekery such as delayed accounting metrics for processes, DragonFly BSD support, improved meter styles for the visual overview of CPU usage, and better battery monitoring on Linux.

Launched just like `top`, and with shortcuts and arguments that offer the same kind of functions, most new users can simply continue with this new tool just as they would with the old one. Firstly, they'll get a much more reactive curses-based interface and overview of what's running. Unlike `top`, there's a function key-driven menu system



`htop` looks even better when you use split `tmux` sessions to monitor a network of machines!

that can be used to change the sort mode, kill processes, and get help. There are lots of other shortcuts, too – such as pressing `c` to see threaded processes or `s` for integrated `strace`. It's quick, powerful, and takes very few resources. If you haven't already switched from `top` to `htop`, version 2.1.0 is the perfect time to do so.

**Project Website**

<http://hisham.hm/htop>

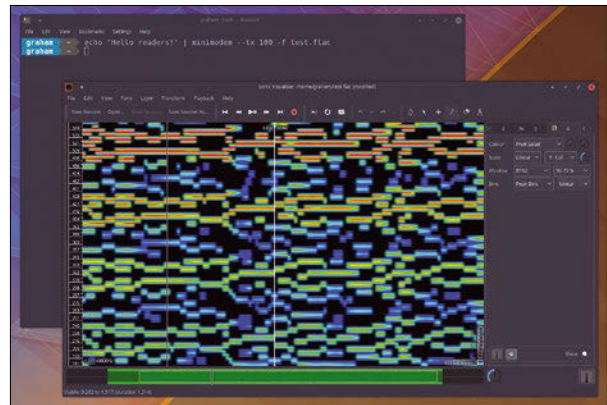
## Software modem

# Minimodem

Even before the Internet was accessible outside of universities, thousands of us wanted to connect our computers to each other. The best solution was to use a modem connected to your serial port to connect to a privately run bulletin board system (BBS), using the likes of FidoNet to communicate across these federated portals, download files, and share updates. The modem was at the heart of this experience, turning the raw data sent from the serial port into an audio stream that could be transmitted across a telephone line. Like loading data from a tape recorder and compact cassette, the sounds emitted by a modem as it negotiated a connection became ingrained into our subconscious. Those

modems may be gone, but the sounds they made can live again – all thanks to Minimodem.

Minimodem can generate those exact modem sounds, directly from the command line. But it's far more than a piece of software nostalgia, as it can even decode those sounds. This makes it a pure software emulation of one of those old hardware units and gives your machine the ability to communicate with any BBS you can still find running, or perhaps that Amiga 1200 you left plugged into the phone line at your parent's place. It's even useful if you have old hardware like the Amiga hanging around, as you can use Minimodem to communicate with a computer that perhaps doesn't have any networking capability but can still talk to a



Even if you don't have a modem with which to send and receive signals, Minimodem generated sound is almost its own music genre.

modem through its serial port. Simply connect an audio cable in lieu of a telephone line and connect the modem at the other end, and you'll be able to launch your own local BBS. Minimodem even lets you turn audio recordings of a session into the data the session would have generated, which may be a great way to reverse engineer WOPR in War Games.

### Project Website

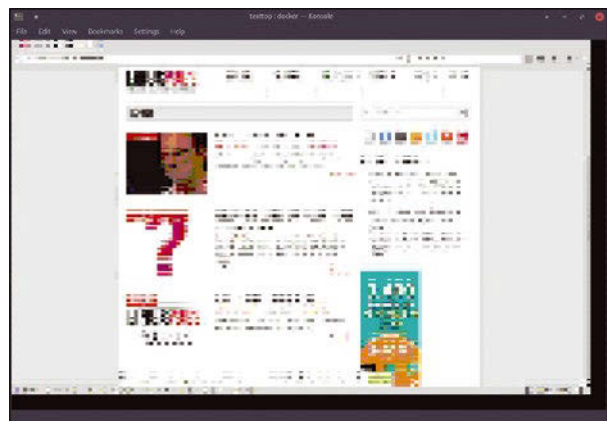
<http://www.whence.com/minimodem/>

## Terminal desktop

# Texttop

The screenshot might look like it's a screen grab from a ZX Spectrum emulator running a copy of Hacker, but it's actually way more advanced than that. It's an X session framebuffer being rendered in a console session, with the framebuffer itself showing Firefox with the *Linux Voice* page open. This kind of thing has been done before, with video or image data being piped through a graphical tool that will turn the imagery into ASCII text. It's possible to do this quickly enough that video input can become a recognizable video output, albeit rendered in text. And because it's text, you can easily view the stream from anywhere with remote terminal access, such as an SSH terminal.

The difference with Texttop is that it's interactive – it's not simply broadcasting a passive stream. This is a big deal because it fills in a missing gap on the command line – the ability to run applications designed for graphical interaction. The web browser is the perfect example, because while command-line alternatives like *links2* are excellent for occasional use, such as downloading new graphics drivers after a kernel update has destroyed your configuration, they're not every good with most websites. For those, you need a fully fledged browser with proper JavaScript, such as Firefox. Texttop lets you access a remote Firefox session through a regular terminal. While it's almost impossible to



Stuck on a low bandwidth connection and need to access a website? Try piping Firefox through Texttop.

use at this stage, the proof of concept does work, and the author is working on a major revision. You can launch a remote browser within Docker and use Mosh to access it via SSH. From there, you can view your remote session and interact with it using the mouse just as you would locally.

### Project Website

<https://github.com/tombh/texttop>

## Storage analyzer

# ncdu 1.13

There are several great ways to check how much storage space you have remaining and where storage is being used. One of the best GUI-driven applications is Filelight as this gives a rapid visual overview of your complete filesystem organized by size. The bigger the file, the more it consumes a pie-chart representation of your directory structure. On the command line, however, you're typically left quite literally to your own devices. Commands like `du` and `dh` can be used with a variety of arguments to present the same kind of data as Filelight, albeit without the color. But there's another option, and that's `ncdu`.

As its name implies, `ncdu` is an ncurses-driven replacement for

`du`, and it's so effective that you're unlikely to want to resort to either a graphical or a command-line alternative. Its main feature is speed. Launch the command and within seconds (storage medium permitting) it will populate a list of files and folders relative to your current location with a hash-character bar graph used to visualize which take up the most space. Pressing `g` switches the graph between this mode and a percentage indicator and then both. Pressing `d` will scarily delete the selected file or directory (after a warning), and pressing `b` will spawn a shell in your new location. All these keyboard shortcuts can be accessed by pressing the `?` key within the utility. But the best thing about all of

Celebrating more than 10 years of updates, the best feature in the new version is the inclusion of color!

this is its speed, as you can quickly skip through the filesystem and instantly see where all the space is being used. It's perfect for a remotely connected Raspberry Pi or any other remote server and is quicker than trying to do the same thing from the desktop.

**Project Website**

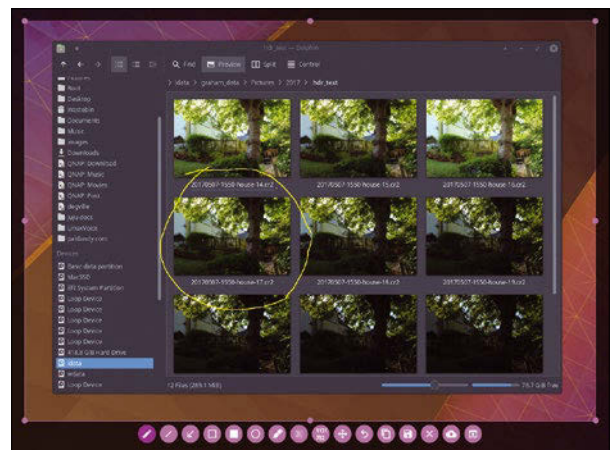
<https://dev.yorhel.nl/ncdu>

## Screenshot taker

# Flameshot

It's amazing that, for a simple tool, the humble screenshot utility hasn't found perfection and ascended to ubiquity years ago. We could all agree that the copy-to-clipboard approach of the `PrntScr` key on Windows needed improvement, as could the name of `Scrot`, but it's been harder to find innovation since `Spectacle` took over from `KSnapshot`. But there's a new contender, and it's gone straight to the top of the pile. `Flameshot` gets nearly everything right, and more importantly, it does this without getting anything wrong. The brilliant `HotShots` gets close, for instance, with its ability to annotate and edit images before they're saved. But saving is convoluted enough to make it a pain to use, despite the advantages.

`Flameshot` avoids these pitfalls and has even innovated over `HotShots`. There's a separate configuration tool for setting things like the GUI colors, the buttons that appear, and exactly how screenshots are named when saved. You can then take screenshots with or without the GUI appearing, but it's the GUI that makes this such a good tool. When `Flameshot` is launched and you select a screen area to grab, that area is immediately highlighted and then augmented by a series of buttons (configured earlier) that let you annotate, sketch, heal, resize, draw lines, and undo any changes you make. You can do all of this before the image is automatically saved or uploaded. The only function it's really missing is the ability to create automatic annotations like you



**Flameshot must be the most comprehensive screen tool to support Gnome Wayland and Plasma Wayland.**

can with `HotShots`, but apart from that, it's perfect. As it's Qt-based rather than KDE-based, it's also desktop agnostic and even supports `Gnome Wayland` and `Plasma Wayland` alongside the usual suspects. If there is one thing it's missing, it's a delay period. Otherwise, it's perfect.

**Project Website**

<https://github.com/lupoDharkael/flameshot>

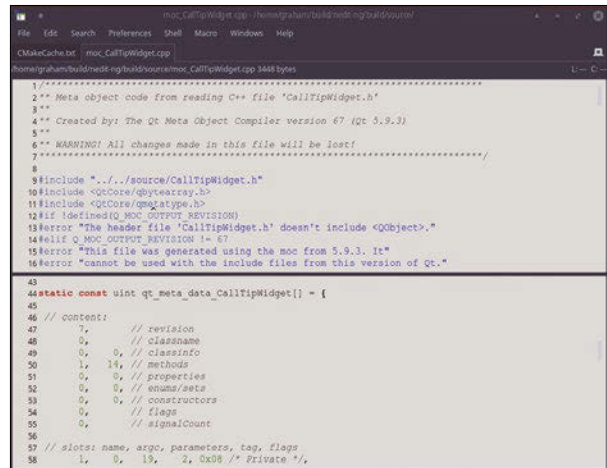
## Text editor

# nedit-ng

This wouldn't be FOSS-Picks if we didn't look at least one text editor, and this month it's nedit-ng's turn. What makes nedit-ng special is that it's a complete and faithful port of the Nirvana Editor (NEdit) v5.6. This is an editor that's been around since the dawn of time, or X windows, depending on which came first, and it still has a popular following because of the austerity of its early-90s user interface and its C-like macro language. The last update to the original version was in 2014, when version 5.6 was released, and it's this version for which nedit-ng is built to be a drop-in replacement. The main reason for this port was to fix some long standing bugs in the original, some of which reportedly

resulted in a crash. But the new editor also updates the old visuals without losing the Motif-like simplicity. It's built using Qt 5, which can look just like Motif if you let it, but by default looks a lot better. It will integrate with your desktop theme, for example, and the colors and syntax highlighting don't look out of place on a modern desktop. The new version even uses anti-aliased fonts.

It's what many people would call a distraction-free editor, because it doesn't waste space with modern additions like an object viewer or dependency manager. It's clean and crisp, with split views, line numbering, simple statistics, and tabs appearing if you need them. There's also some excellent



```

1 //*****
2 ** Meta object code from reading C++ file "CallTipWidget.h"
3 **
4 ** Created by: The Qt Meta Object Compiler version 67 (Qt 5.9.3)
5 **
6 ** WARNING! All changes made in this file will be lost!
7 //*****
8
9 #include "../source/CallTipWidget.h"
10 #include <QtCore/qbytearray.h>
11 #include <QtCore/qstring.h>
12 #if !defined(Q_MOC_OUTPUT_REVISION)
13 #error "The header file 'CallTipWidget.h' doesn't include <QObject>."
14 #elif Q_MOC_OUTPUT_REVISION != 67
15 #error "This file was generated using the moc from 5.9.3. It"
16 #error "cannot be used with the include files from this version of Qt."
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43 #static const uint qt_meta_data_CallTipWidget[] = {
44
45 // content:
46
47 7, // revision
48 0, // classname
49 0, // classinfo
50 1, 14, // methods
51 0, 0, // properties
52 0, 0, // enums/sets
53 0, 0, // constructors
54 0, // flags
55 0, // signalCount
56
57 // slots: name, argc, parameters, tag, flags
58 1, 0, 19, 2, 0x00 /* Private */,

```

**Edit text like it's 1992 with the complete rewrite of the Nirvana Editor.**

syntax highlighting to make your code easier to read. In fact, the syntax highlighting is the product of some carefully crafted regular expressions, which means you can edit them yourself if you've got the skills. It's just like moving from Vi to Vim, which must be exactly what the authors planned.

**Project Website**

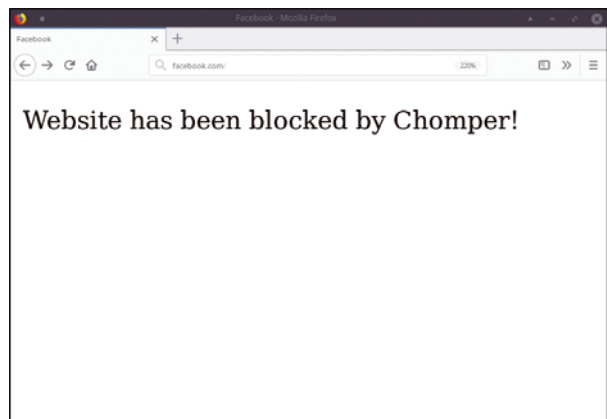
<https://github.com/eteran/nedit-ng>

## Internet blocker

# Chomper

This sentence is being typed with the help of two random Internet searches, one YouTube video intermission, and a brief chat with a remote acquaintance. All thanks to reasonably fast and cheap broadband. And this is how most sentences are now being written, and not just on these pages. One side effect of pervasive Internet that most of us could never have anticipated is its ability to suck the life out of attention spans – thirty seconds work, thirty seconds checking social networks. But it doesn't have to be this way, and like any good food diet, cutting back on fatty food or distractions is most effective when there's some third-party oversight. Chomper is that oversight.

Chomper is basically a Python script that blocks access to a group of Internet sites taken from a blacklist. It can also enable access for only a specific amount of time, and mix up times, blacklists, and whitelists according to the profile you want to use for whatever length of time you want to use it. All of this is done through an easy-to-understand YAML configuration file. Create a social profile and block Facebook, for example, or just enable access to your GitHub page. It works far better than a dynamically updated hosts file, which often accomplishes little more than hiding the cookies in a different drawer. The project's readme also offers excellent advice for Linux users with root access for enabling



**Go on an Internet diet and get your productive life back with Chomper.**

“hardcore mode.” This is really just good advice for running Chomper – set up an admin account, give it an impossibly long password, lock this away somewhere on paper, and give this account permission to run the script. With that done, you won't be able to stop Chomper until the time is up, and that's when Chomper works best.

**Project Website**

<https://github.com/aniketpanjwani/chomper>



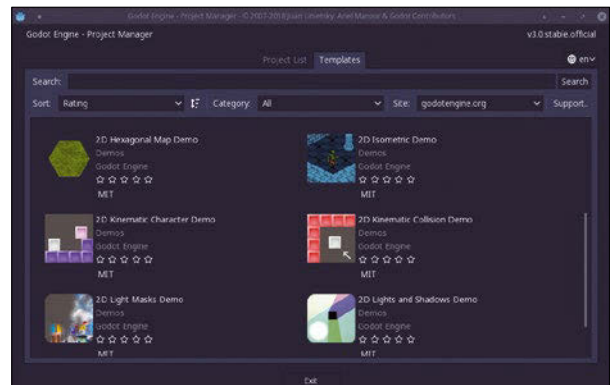
## Game engine

# Godot 3.0

In recent years, Godot has gone from being an interesting side project for people wanting to write their own open source games, to a serious alternative to game engines like Unity and Unreal. This is important because the games industry, in particular, is still driven by proprietary games written on proprietary platforms. While there's absolutely nothing wrong with this, having a "free software" platform that attempts to implement a similar system is wonderful for all kinds of reasons. It means, for example, that educators can teach games development without the cost associated with platform licensing, or even worse, being restricted in how the games created with education versions can be used or sold. It means students have no cost barrier for entry, and it means game developers who want to release every line and asset of their creation can do so without complicating the situation by building atop a "free" platform. Godot has grown to fill this purpose, much like Blender or Ardour have in their respective

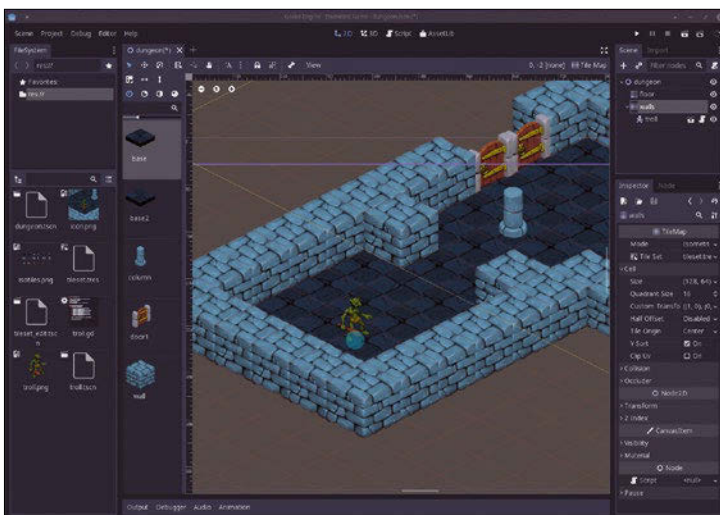
realms, and Godot 3.0 is by far its biggest release yet.

Godot includes an editor, both 2D and 3D games engines, and animation tools, and it's completely open source and cross platform. This major new release has taken 18 months to get to this point. During this time, the community surrounding the project grew to such an extent that its developers created a Software Freedom Conservancy-ratified Patreon page, which has in turn funded the full-time employment of one of its core developers alongside an additional developer to work on a new rendering backend and the brilliant new GDNative for this release. GDNative is a way for Godot to dynamically use external libraries without recompiling the engine. It means, for example, you can dynamically support various audio and video codecs, and build games for virtual reality or Microsoft's Kinect without having to build a special binary for each use case. GDNative also allows for the use of extra scripting languages, such as the very popular (in gaming circles) Mono/C#.



**Download open source/free software assets and examples directly within Godot 3.0.**

This new version feels like a completely different piece of software. Elements such as the integrated asset download tool and asset workflow (complete with licence listing) and a fantastic set of new documentation help make this feel like commercial software; there are too many new features to list. Perhaps the strongest new feature is the new 3D renderer, which includes the full principled BSDF shader. We covered the details of this fantastic shader when it was added to Blender, and the results are equally stunning here. Alongside this comes global illumination with a mode called GI that probes for real-time lighting and a classical lightmapping workflow for slower machines. The end results are a major upgrade of Godot's appearance and form a cornerstone of the new version being used for professional quality production. New tone mapping, GPU particles, and support for GLTF 2.0 (the latest standard of file formats for 3D scenes) close the deal. The end result: If you've been looking to get into Linux game development but thought there was nothing to compete with the likes of Unity, Godot is now worth your time and money.



There has never been a better time to get into game development, and you can do it for free with Godot and its huge community of supporters.

**Project Website**  
<https://godotengine.org>

# Location, Location, Location

Even though your computer does not come with its own GPS, you can hack one onto it using a mobile phone. Although, it does take quite a bit of fiddling.

BY PAUL BROWN

There is plenty of literature online on how to connect a GPS device to a computer. But normally the devices covered are exclusively GPSs, that is, something you would probably have to go out of your way to buy.

What you don't find so much information about is how to use what you have at hand, namely, your regular phone. However, it can be done. It is a bit hacky, but it works.

## On Your Laptop

Linux comes with a daemon called `gpsd` [1] that can process the information delivered by a GPS device. Although `gpsd` is usually started and stopped with `systemd` (and is usually launched at boot), you'll be running it by hand. The reason is

### NMEA

GPS devices deliver information in the shape of National Marine Electronics Association (NMEA) packages. These packages that conform to the NMEA's 0183 or 2000 standards contain all the data you need to position a GPS device on the Earth's surface.

NMEA packages contain "sentences" that look like this:

```
GPRMC,092751.000,A,5321.6802,N,00630.3371,W,0.06,31.66,280511,0.0,A
```

An NMEA sentence can contain data on the device's altitude, latitude, longitude, and speed. It also indicates the time of the reading at the beginning of the sentence and a checksum at the end. This helps identify broken or partially received sentences.

Most programs and programming libraries used for positioning parse NMEA packages and sentences automatically. This means you usually won't need to write any code to turn the raw NMEA data into something usable.

again the hackiness of the solution, due to processing data coming from a GPS device not on board or that is not connected over a serial or USB port, which is what `gpsd` usually expects. (See the "NMEA" box.)

`gpsd` is available in most, if not all, distributions. On Debian and its derivatives, it is installed with:

```
apt-get install gpsd
```

If you are on Ubuntu, Raspbian, or something similar, you may have to add `sudo` to the preceding instruction.

Be that as it may, this command will install (and run) the GPS daemon and install several utilities such as `gpsmon` and `gpspipe`.

The first thing you want to do is stop the `systemd` `gpsd` services. As explained previously, you want to run `gpsd` with your own parameters. You can later work your special configuration into the `systemd` version, but for now, run these two instructions to free up the `gpsd` service and the port on which it listens.

```
systemctl stop gpsd
systemctl stop gpsd.socket
```

If this doesn't do the trick, check what `systemd` is running with:

```
systemctl | grep gpsd
```

and stop whichever service shows up there.

You may have to remove the `/var/run/gpsd.socket` file as well (you will have to do that with `sudo`) if you are going to run `gpsd` as a regular user.

## On Your Phone

On your phone, you need an app that shares your GPS data over a network. There are some quite fancy proprietary programs out there, but, unless you go for the "premium" version, they are usually addled with ads and bloat.

Fortunately, there is a very simple option created by Tiago Koji Castro Shibata that you can get from the Google Play app store [2] or download and compile yourself from the author's GitHub page [3]. When you run it, it looks like Figure 1.

Configuring the app is simplicity itself: Input your laptop's IP. Choose a port to which to forward the information. Press the Start button. The end.

### Connecting Both

Back on your laptop, check that your IP corresponds to what you input on the mobile app and run the `gpsd` daemon from a terminal window:

```
gpsd -N upd://*:<port>
```

What you are doing here is polling information from a phone, any phone, on the port you established on the app. To all practical effects, your phone is acting like a server.

For example, if you are connecting to an app configured as shown in Figure 1, you would run:

```
gpsd -N upd://*:29998
```

The `-N` parameter lets you run `gpsd` without it going into the background.

Once you are happy that everything works as it should, you can run `gpsd` without it.

As for why “any” phone and not your own specific phone, that's because, in theory you could select a specific source on the network by passing your phone's IP to `gpsd`, like so:

```
gpsd -N udp://<phone's IP>:<port>
```

However this did not work for me. The only way `gpsd` managed to pick up the server on our phone was by using the catch-all `*`. This shouldn't be a big problem: Pick an atypical and unique port, and it is

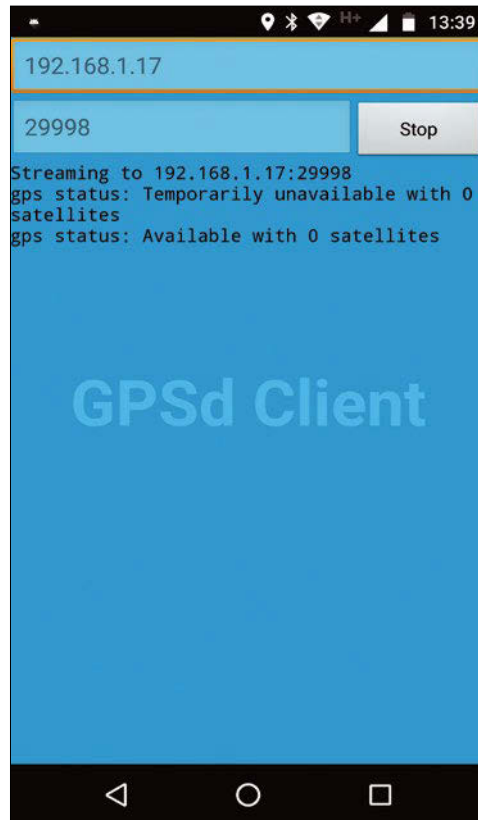


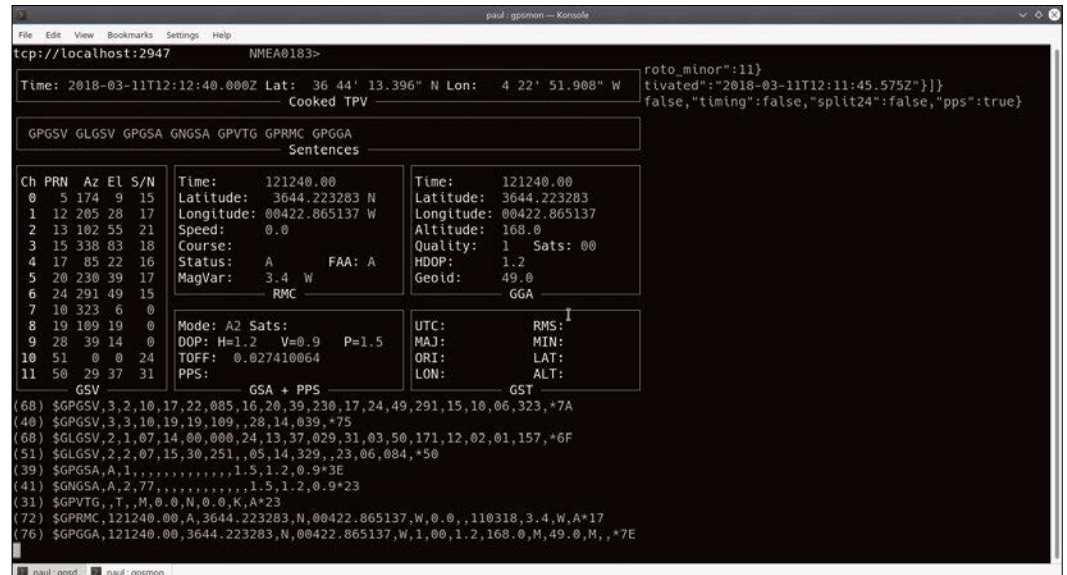
Figure 1: A simple GPS client that allows you to share data with your computer.

unlikely that `gpsd` will poll anything that isn't the device you want it to poll.

Now move around with your phone, preferably near a window, to try and capture some coordinates. Then open another terminal window and run `gpsmon`.

The `gpsmon` utility does what it says on the box: It monitors GPS input. Without any parameters, it monitors what `gpsd` is receiving, but it can also monitor a stream of GPS data from other sources, including the Internet. Its text-based interface shows information on the satellites in view (the `GSV` section in Figure 2), data which provides 3D location and accuracy (the `GGA` section), and so on.

Figure 2: The `gpsmon` utility shows you GPS in action.



A healthy GPS connection will look like Figure 2. If you see something different, wait a while to see if the stream starts, or move your phone to a location where it can “see” satellites, near or outside a window. If that doesn't work, restart `gpsd`, `gpsmon`, the app on your phone, or all three. GPS is a fickle beast and breaks a lot. If you have followed the instructions above, at some point you will get a signal, and the stream of data will begin.

This is all very well if you are at home and running stuff over the local network, but you would

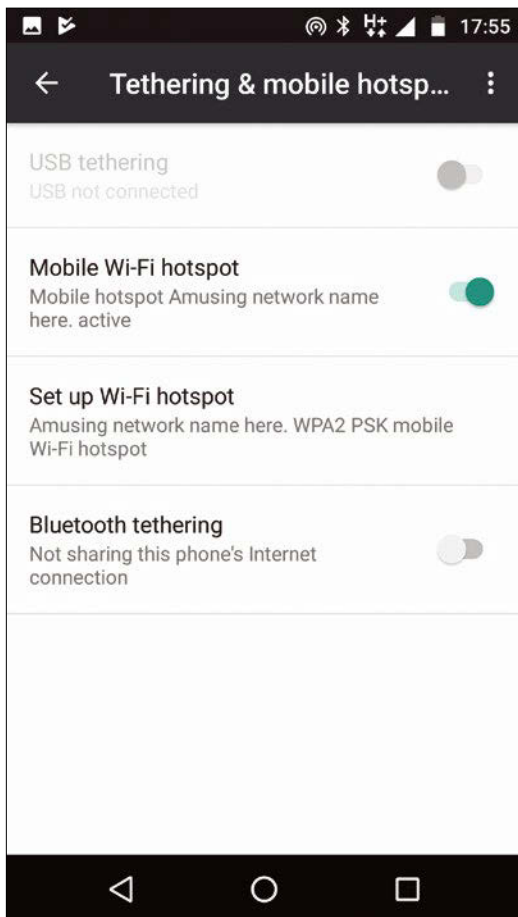


Figure 3: Activate the *Mobile Wi-Fi hotspot* feature on your phone and take your GPS rig on the road.

usually want to use your GPS-enabled system when you are out and about. What then? Use your phone as a WiFi hotspot (Figure 3)! Just because your phone is the hotspot, it doesn't mean it cannot stream GPS data to your laptop.

Activate *Mobile Wi-Fi hotspot* from *Settings | Wireless & networks | More | Tethering & mobile hotspot* on your phone, connect to your phone's network from your laptop, and your computer will be assigned an IP that is something like `192.168.43.X` (for the record, phones usually adopt `192.168.43.1` as their IP when acting as a hotspot). Use that with your Android GPS client, and you will be able to take your GPS experiment out and about.

A final `gpsd` tool to be aware of is `gpspipe`. At

its simplest, `gpspipe` takes a parameter that tells it which format to use and just dumps the stream from `gpsd` to the standard output:

```
gpspipe -r
```

will stream raw NMEA sentences to the terminal, while

```
gpspipe -w
```

sends native `gpsd` sentences (see Listing 1).

`gpspipe` can also output to a file using the `-o` parameter, be sent to the background as a daemon (`-d`), or send data to a serial device (`-s`).

You could use `gpspipe` to send all the information to a file and keep track of a trip or a run. In the

case of a run, you could also see the terrain's profile and figure out how much you climbed, since the NMEA sentences also include the altitude at any given point in your route.

I could end the article right here. You have connected your phone; you can use its GPS on your laptop or on a single board computer such as the Raspberry Pi; and you have learned how to store the data. With that alone, you can carry out a bunch of cool projects. None of which, useful as they may be, will be very visually appealing.

So let's make a graphical map-tracking application instead.

### Mapping Out

The objective is to show a map of the GPS device's location, and, as it moves, have the map move with it (see Figure 4). A circle indicates where the device is located on the map.

Your first reaction would probably be to base your app on some open source map app that already exists. Marble [4] seems like a good candidate, and if you run it like so

```
marble --latlon <lat>,<lon> --distance 2
--map "earth/openstreetmap/openstreetmap.dgml"
```

where `lat` and `lon` are the latitude and longitude, Marble will open at the desired location, which is already promising.

Then, if you think that

```
gpspipe -w | sed -n s/.*lat:(.*/),/1/p |
cut -d, -f1
```

will give you the latitude of your phone, and

```
gpspipe -w | sed -n s/.*lon:(.*/),/1/p |
cut -d, -f1
```

will give you the longitude, things definitely seem to be shaping up.

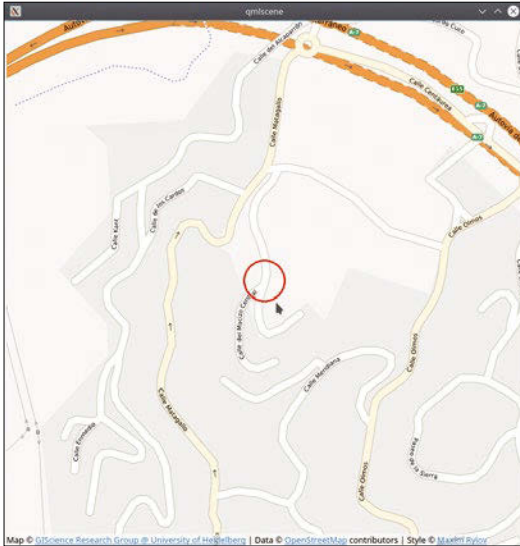
The preceding instructions will give you an ongoing stream of latitudes and longitudes. It would be possible to pipe the output to a file and then run through the rows and use them with Marble somehow. But this is an inelegant and convoluted way of doing things. It is much smarter to break out QML [5].

### Listing 1: `gpspipe -w` output

```
{ "class": "TPV", "device": "udp://*:29998", "mode": 3, "time": "2018-03-11T09:21:21.000Z", "ept": 0.005, "lat": 36.736763683, "lon": -4.381267583, "alt": 149.000, "epv": 20.700, "track": 0.0000, "speed": 0.000 }
{ "class": "SKY", "device": "udp://*:29998", "vdop": 0.90, "hdop": 1.50, "pdop": 1.70, "satellites": [ { "PRN": 7, "e1": 26, "az": 45, "ss": 45, "used": false }, { "PRN": 13, "e1": 45, "az": 298, "ss": 24, "used": false }, { "PRN": 28, "e1": 35, "az": 113, "ss": 18, "used": false }, { "PRN": 30, "e1": 51, "az": 46, "ss": 38, "used": false },
...
] }
```

## QML App

QML is a relatively simple declarative language that uses the Qt libraries to build graphical applications. How QML works is a bit like how you'd go about programming an interface using a visual IDE, but with text: You "design" the layout de-



**Figure 4:** Your app will show the position of your phone on a map. As it moves, the red circle will follow it around.

scribing it in text form, and then you do the same with all the items contained in the layout. To execute actions (say, when a button is pressed), you associate events with JavaScript modules to each object.

Getting back to the original idea, the Marble documentation says that it is easy to develop apps based on the Marble code in QML. It refers the would-be programmer to a tutorial and examples available at [6]. Unfortunately, the code examples and snippets are outdated, and it is quite hard to figure out how to do things. According to one developer, the problem is that Marble's code is in continuous flux, changing nearly daily.

Instead, he suggests using QML's own *Location* library. To get started, install all the bits and pieces you'll need:

```
apt install qmlscene qml-module-qtpositioning 2
qml-module-qtlocation
```

The modules will install positioning and location libraries for your app and drag in all the Qt QML common files. The `qmlscene` utility is what you use to run QML apps.

### Listing 2: map.qml

```
01 import QtQuick 2.1
02 import QtQuick.Window 2.0
03 import QtPositioning 5.5
04 import QtLocation 5.6
05
06 Window {
07     id:page
08     width: 800
09     height: 800
10     visible: true
11
12     Map {
13         id:myMap
14         anchors.fill: parent
15         plugin: mapPlugin
16         zoomLevel: 17
17
18         property MapCircle circle
19
20         function update(pos) {
21             removeMapItem(circle);
22
23             circle = Qt.createQmlObject('import QtLocation 5.3;
24                                     MapCircle {}', page);
25             circle.radius = 30;
26             circle.color = "transparent";
27             circle.border.color = "red"
28
29             circle.border.width = 3;
30             myMap.addMapItem(circle);
31
32             circle.center = pos.coordinate;
33             myMap.center = pos.coordinate;
34
35             //console.log("Coordinates: ", pos.coordinate.
36                             latitude, pos.coordinate.longitude);
37         }
38     }
39
40     Plugin {
41         id: mapPlugin
42         name: "osm"
43     }
44
45     PositionSource {
46         id: gpsPos
47         updateInterval: 500
48         active: true
49         nmeaSource: "socket://localhost:29999"
50
51         onPositionChanged: {
52             myMap.update(position);
53         }
54     }
55 }
```

The code for your app might look like Listing 2.

After loading in all the modules you need (lines 1-4), you create an all-encompassing `Window` object (line 6). You give it a name (`page`), set its size, and make it visible.

Next you create a `Map` widget. Again, you give it a name (`myMap`), you set its size to automatically fit in `Window` (the `parent` object), and you set the plugin it will use. In this case, the plugin is declared on lines 37 to 40, where you can see you will be using the `OpenStreetMap` [7] data for your maps. Finally, you set the zoom level of the map, in this case, 17 meters up in the air.

`Map` objects allow you to draw shapes on the map. If you want a circle, you create a `circle` object on line 18. You then manipulate it using a JavaScript function (lines 20-34) and moving it (line 30) when the GPS position changes. You also move the view of the map so that the circle is always in the center (line 31).

Line 33 is for debugging and testing only. Uncomment it to see coordinates scroll by on the terminal window as you move around.

As mentioned previously, QML is declarative and event-based and you can see how that works in the `PositionSource` object (lines 42-49). This is an object provided by the `QtPositioning` module, and it is in charge of retrieving and processing the GPS information.

As usual, you give your object a name (`gpsPos`) to refer to it later, you set how often it will poll the GPS source (once every 500 milliseconds), and you set from where it will get the NMEA information.

A slight detour here, because it turns out that, unless you have GPS on the laptop or a GPS-only device connected via USB (that is, *not* a phone), `PositionSource` does a poor job of determining where it should get its data.

However, you can force it to look for data by setting the `nmeaSource` attribute. You would usually `nmeaSource` to point to a file with static NMEA data to “simulate” a GPS positioning (`nmeaSource` is primarily used when testing), but you can also point to a server. This got me thinking: First I tried to pipe the output from `gpspipe` to a file and read from that:

```
gpspipe -r > gps.dat
```

Line 46 then looked like this:

```
nmeaSource: "gps.dat"
```

But then `PositionSource` was never updated – it just took the last value from the file and never went back to check if there were any more.

A bit more research revealed that `PositionSource` can also be read from a socket URL. And

know what makes sockets super easy? Netcat [8] does. Netcat (`nc`) is preinstalled in nearly every Linux installation on the planet, so it was simply a matter of doing this:

```
gpspipe -r | nc -l 29999
```

Then, create a stream of GPS data on `localhost` using port 29999, and get `PositionSource` to read from it, as shown on line 46.

Finally, you set up an event that fires when the position you are polling changes (line 48). When it does fire, you visit the `update(pos)` function in the `Map` object (lines 20-34) and redraw the circle and the map with the new position data, `pos.coordinate`, that you pass in the function call.

To summarize, here’s what you do to make everything work together:

1. Run `gpsd udp://*.29998`.
2. Connect your phone’s GPS client to your laptop using your laptop’s IP and port 29998.
3. Run `gpsmon` to check you are receiving GPS data correctly.
4. Run `gpspipe -r | nc -l 29999` to set up a data stream through port 29999 on your computer.
5. Run your application with `qmlscene map.qml`.

## Conclusion

There you have it: A full GPS app on a laptop (or Raspberry Pi) in 50 lines of code [9]. This is definitely a fun project, and it combines hardware you probably already have hanging around at home in an interesting way. Expanding to build an actually useful application should be a piece of cake.

Famous last words. ■■■

## Info

- [1] `gpsd`: <http://www.catb.org/gpsd/>
- [2] `gpsd` client on Google Play: <https://play.google.com/store/apps/details?id=io.github.tiagoshibata.gpsdclient>
- [3] A (free) `gpsd` client for Android: <https://github.com/tiagoshibata/Android-GPSd-Client>
- [4] Marble: <https://marble.kde.org/>
- [5] QML: <https://doc.qt.io/qt-5.10/qtqml-index.html>
- [6] Marble for developers: <https://marble.kde.org/dev-intro.php>
- [7] OpenStreetMap project: <https://www.openstreetmap.org/>
- [8] Netcat: <http://nc110.sourceforge.net/>
- [9] Video of the app in action: <https://youtu.be/jcHOY07Rvpk>

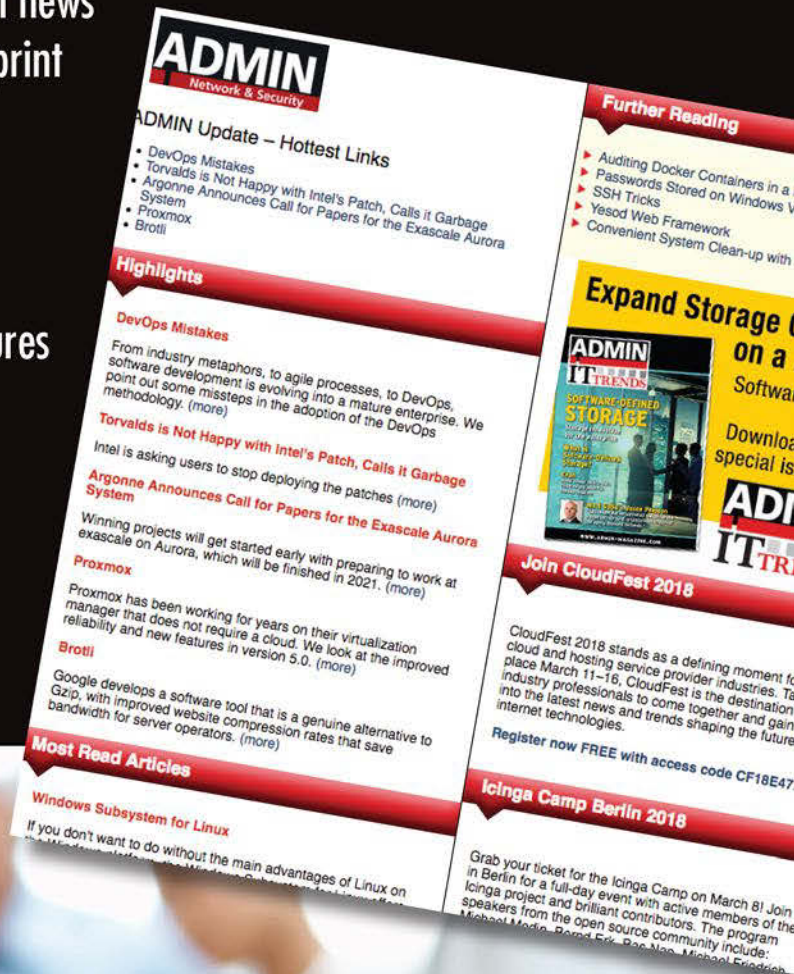
# Too Swamped to Surf?

Our ADMIN Online website offers additional news and technical articles you won't see in our print magazines.

Subscribe today to our free ADMIN Update newsletter and receive:

- Helpful updates on our best online features
- Timely discounts and special bonuses available only to newsletter readers
- Deep knowledge of the new IT

**ADMIN**  
Network & Security



© Yann Arcurs, fotolia.com

[www.admin-magazine.com/newsletter](http://www.admin-magazine.com/newsletter)

# 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](mailto:events@linux-magazine.com).



## openSUSE Conference 2018

**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 meet ups and hack sessions.

## Texas Linux Fest

**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.

## ISC High Performance 2018

**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

heise Security Tour 2018	April 10, 12, 18, 24, and 26, 2018	Numerous European cities	<a href="https://www.heise-events.de/securitytour">https://www.heise-events.de/securitytour</a>
International Javascript Conference 2018	April 11–13, 2018	London, England	<a href="https://javascript-conference.com/program/">https://javascript-conference.com/program/</a>
2018 HPC for Wall Street	April 16, 2018	New York, New York	<a href="http://www.flagmgmt.com/linux/">http://www.flagmgmt.com/linux/</a>
Cloud Foundry Summit North America	April 18–20, 2018	Boston, Massachusetts	<a href="https://www.cloudfoundry.org/event/nasummit2018/">https://www.cloudfoundry.org/event/nasummit2018/</a>
Linux Presentation Day 2018.1	April 21, 2018	Europe-wide in many cities	<a href="http://www.linux-presentation-day.org/">http://www.linux-presentation-day.org/</a>
Linux Storage Filesystem and Memory Management Summit	April 23–25, 2018	Park City, Utah	<a href="http://events.linuxfoundation.org/events/lfmm-2018/">http://events.linuxfoundation.org/events/lfmm-2018/</a>
LinuxFest Northwest	April 28–29, 2018	Bellingham, Washington	<a href="https://linuxfestnorthwest.org/conferences/lnfw18">https://linuxfestnorthwest.org/conferences/lnfw18</a>
KubeCon + CloudNativeCon	May 2–4, 2018	Copenhagen, Denmark	<a href="https://events.linuxfoundation.org/events/kubecon-cloudnativecon-europe-2018/">https://events.linuxfoundation.org/events/kubecon-cloudnativecon-europe-2018/</a>
openSUSE Conference 2018	May 25–27, 2018	Prague, Czech Republic	<a href="https://events.opensuse.org/conference/oSC18">https://events.opensuse.org/conference/oSC18</a>
Texas Linux Fest	June 8–9, 2018	Austin, Texas	<a href="https://2018.texaslinuxfest.org/">https://2018.texaslinuxfest.org/</a>
CeBIT 2018	June 11–15, 2018	Hannover, Germany	<a href="http://www.cebit.de/en/">http://www.cebit.de/en/</a>
Open Source Data Center Conf.	June 12–13, 2018	Berlin, Germany	<a href="https://osdc.de/">https://osdc.de/</a>
ISC High Performance 2018	June 24–28, 2018	Frankfurt, Germany	<a href="http://www.isc-hpc.com/">http://www.isc-hpc.com/</a>
DebConf 18	July 29–August 5, 2018	Taipeh, Taiwan	<a href="https://wiki.debconf.org/wiki/DebConf18">https://wiki.debconf.org/wiki/DebConf18</a>
Akademy 2018	August 11–17, 2018	Vienna, Austria	<a href="https://akademy.kde.org/2018">https://akademy.kde.org/2018</a>
DevOpsDays Berlin	September 12–13, 2018	Berlin, Germany	<a href="https://www.devopsdays.org/events/2018-berlin/welcome/">https://www.devopsdays.org/events/2018-berlin/welcome/</a>



# 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](mailto:edit@linux-magazine.com).



## Authors

Erik Bärwaldt	16, 22, 64
Swapnil Bhartiya	8
Paul Brown	90
Zack Brown	12
Bruce Byfield	34, 60
Andy Carlson	52
Joe Casad	3
Mark Crutch	69
Alexander Fillbrunn	46
Jon "maddog" Hall	75
Martin Horn	46
Charly Kühnast	45
Christoph Langner	42
Vincent Mealing	69
Martin Mohr	56
Andreas Möller	30
Graham Morrison	84
Hartmut Noack	80
Dmitri Popov	70
Mike Schilli	38
Ferdinand Thommes	76

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](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.

## Contact Info

### Editor in Chief

Joe Casad, [jcasad@linux-magazine.com](mailto:jcasad@linux-magazine.com)

### Managing Editor

Rita L Sooby, [rsooby@linux-magazine.com](mailto:rsooby@linux-magazine.com)

### Localization & Translation

Ian Travis

### News Editor

Swapnil Bhartiya

### Copy Editor

Amy Pettle

### Layout

Dena Friesen, Lori White

### Cover Design

Lori White

### Cover Image

© Vasily Pindyurin, 123RF.com

### Advertising

Brian Osborn, [bosborn@linuxnewmedia.com](mailto:bosborn@linuxnewmedia.com)  
phone +49 89 3090 5128

### Marketing Communications

Gwen Clark, [gclark@linuxnewmedia.com](mailto:gclark@linuxnewmedia.com)  
Linux New Media USA, LLC  
616 Kentucky St.  
Lawrence, KS 66044 USA

### Publisher

Brian Osborn

### Customer Service / Subscription

For USA and Canada:  
Email: [cs@linuxpromagazine.com](mailto:cs@linuxpromagazine.com)  
Phone: 1-866-247-2802  
(Toll Free from the US and Canada)  
Fax: 1-785-856-3084

For all other countries:  
Email: [subs@linux-magazine.com](mailto:subs@linux-magazine.com)

[www.linuxpromagazine.com](http://www.linuxpromagazine.com) – North America

[www.linux-magazine.com](http://www.linux-magazine.com) – Worldwide

While every care has been taken in the content of the magazine, the publishers cannot be held responsible for the accuracy of the information contained within it or any consequences arising from the use of it. The use of the disc provided with the magazine or any material provided on it is at your own risk.

Copyright and Trademarks © 2018 Linux New Media USA, LLC.

No material may be reproduced in any form whatsoever in whole or in part without the written permission of the publishers. It is assumed that all correspondence sent, for example, letters, email, faxes, photographs, articles, drawings, are supplied for publication or license to third parties on a non-exclusive worldwide basis by Linux New Media USA, LLC, unless otherwise stated in writing.

Linux is a trademark of Linus Torvalds.

All brand or product names are trademarks of their respective owners. Contact us if we haven't credited your copyright; we will always correct any oversight.

Printed in Nuremberg, Germany by hofmann infocom GmbH on recycled paper from 100% post-consumer waste; no chlorine bleach is used in the production process.

Distributed by Seymour Distribution Ltd, United Kingdom

LINUX PRO MAGAZINE (ISSN 1752-9050) is published monthly by Linux New Media USA, LLC, 616 Kentucky St., Lawrence, KS, 66044, USA. Periodicals Postage paid at Lawrence, KS and additional mailing offices. Ride-Along Enclosed. POSTMASTER: Please send address changes to Linux Pro Magazine, 616 Kentucky St., Lawrence, KS 66044, USA.

Published monthly in Europe as Linux Magazine (ISSN 1471-5678) by: Sparkhaus Media GmbH, Ziebländstr. 1, 80799 Munich, Germany.

Issue 211 / June 2018

# Mycroft

Shopping for a voice assistant is like so many other things in the high tech industry – a loss of privacy is often part of the purchase price. If you really want to explore the realm of voice technology, but you're not interested in letting Google and Microsoft data-mine your home life, try Mycroft – the open source voice assistant.

**Approximate**

UK / Europe

May 05

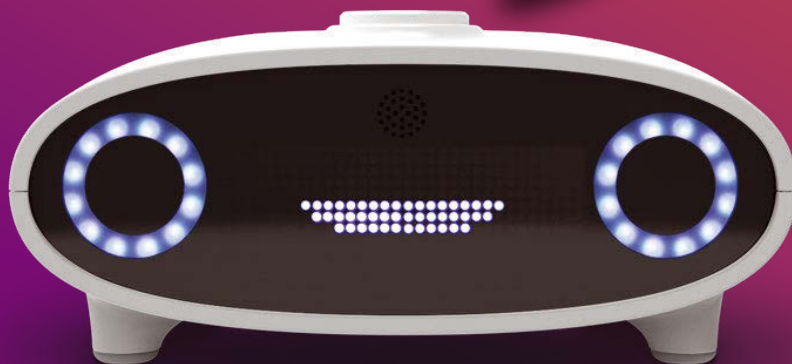
USA / Canada

Jun 01

Australia

Jul 02

**On Sale Date**



## 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](http://www.linux-magazine.com/newsletter)

# ISC HIGH PERFORMANCE 2018

FRANKFURT, GERMANY

- 5 DAYS
- 450 SPEAKERS
- 150 EXHIBITORS
- 3500 ATTENDEES



SAVE THE DATE!



Platinum Sponsors



Beyond Moore's Law | Exascale Systems | Climate Change | HPC and Electric Power Grid Control |  
Big Data Analytics | Cosmology and HPC | Human Brain Modeling and Related Big Data Challenges |  
Future Applications for Quantum Computers | Robotics | What's New with Cloud Computing for HPC? |  
Future Challenges for Programming Models and Languages | The Rise of Containerized HPC |  
Artificial Intelligence on HPC Platforms

[isc-hpc.com](http://isc-hpc.com)

# Application-Optimized Server Solutions

Exactly the Best Performance & Density at Enterprise Scale



## The Most Efficient Data Center in the World

A Fortune 100 Enterprise deploys over 50,000 Supermicro® MicroBlade™ servers at its Silicon Valley data center. With a PUE of 1.06, and Intel® Xeon® Scalable Processors it achieves an 88% improvement in energy efficiency and over \$13M in annual savings. For Exactly the Best Enterprise Computing Solution for your Application, Choose Supermicro.



*Intel Inside®. Powerful Productivity Outside.*



Application-Optimized Server Solutions

*Learn more at [supermicro.com](http://supermicro.com)*