

FREE DVD

fedora

29

Ubuntu 18.10
Desktop (64-bit)

Double-Sided DVD
INSIDE!

TIPS FOR USING
CASCADING
STYLE SHEETS

NAVIGATION
TOOLS FOR EXTENDING
OPENSTREETMAP

LINUX
MAGAZINE

LINUX **PRO**
MAGAZINE

JANUARY 2019

NAVIGATION

Free services that extend
OpenStreetMap

Kernel Disposal

Are old kernels sucking
up valuable space on your
hard drive?

Build a Robot Car

Zorin OS

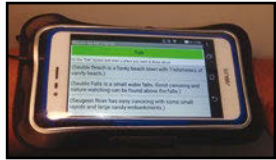
Will this well-appointed
Linux succeed in attracting
Windows users?

WebAuthn

Authenticate
without a password

OOM Killer

What happens when the
kernel runs out of memory?



Put It On!

Build your own wearable with
Android and MIT's App Inventor



LINUXVOICE

- **Retrosahre** – Secure communication over the Internet
- **maddog** – IoT in Brazil
- **Military-Malware Complex** – Security, cyberwar, and the search for zero days



FOSSPicks

- Whalebird
- musikcube
- Imaginary Teleprompter

Tutorial

Getting started with
KDE's Plasma desktop

Issue 218
Jan 2019
US\$ 15.99
CAN\$ 17.99



WWW.LINUXPROMAGAZINE.COM

TRAFFIC WITHOUT LIMITS AND FREE OF CHARGE
FOR DEDICATED ROOT AND MANAGED SERVERS



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
- ✓ Traffic unlimited
- ✓ No minimum contract
- ✓ Setup Fee \$138

Managed Server MX92

- ✓ Intel® Core™ i7-4770 Quad-Core incl. Hyper-Threading Technology
- ✓ 32 GB DDR3 RAM
- ✓ 2 x 4 TB SATA 6 Gb/s 7200 rpm HDD Class Enterprise
- ✓ 100 GB Backup Space
- ✓ Traffic unlimited
- ✓ No minimum contract
- ✓ Setup Fee \$103.50



monthly from \$ **63.00**

monthly \$ **103.50**

www.hetzner.com

SIX HONEST SERVING MEN

Dear Reader,

I must admit I'm prone to writing about the Internet in this space. The Internet is a big place, and there is lots to cover. I often talk about privacy and the financial forces that are shaping Internet trends. Today, though, I'm moved to consider all the weird, creepy lies that clutter up Internet blogs and so-called "news" sites. It's not that I have a solution, but I still think I might have some insights that are worthy of 700 words.

Journalism was once built upon the foundation of what teachers used to call "the five Ws": *who, what, when, where, and why*. Another word, "how," which didn't make the list because it didn't start with W, is also important in some types of news stories. The whole collection is best summed up in a ditty by Imperial British author Rudyard Kipling:

*I keep six honest serving men (they taught me all I knew);
Their names are What and Why and When and How and
Where and Who.*

The rest of the poem, which you can easily find online, is actually a bit more complicated and ambiguous than this pithy quatrain, but I won't go into it all, because this a welcome column, not an English essay. The point is that several generations of journalists who sought to practice their craft the right way have commandeered this passage for inspiration.

Careful attention to the five Ws, or to Kipling's 5W+H formulation, keeps a news story grounded and clear. But it often takes a lot of effort to trace down these details. The economies of our modern technology are such that many authors just don't bother. The result is that news stories have gotten a lot more vague. And the increased tolerance for vagueness among the readership is what gives conspiracy theorists and fake news artists a way into the reader's mind.

The classic "news analysts" you hear on cable networks just want to talk about "why" and "how" without bothering with the rest, and the result is that they harangue each other endlessly from separate universes because they have no shared understanding of the facts.

The classic conspiratorial tweet blurts out some dark pronouncement ("what") or singles out an individual for attack ("who") in a way that causes much more fear and over-reaction than if the author had taken the trouble to fill in the rest of the picture.

For instance, an angry headline might say that the US is being invaded by a rogue caravan of illegal immigrants streaming meth and mayhem in their wake, when a more studious approach that pays heed to Kipling's serving men would yield up something like the following:

- What: a large group of around 5000 migrants heading north through Mexico

- Who: men, women, and children
- When: the latest wave left Honduras on October 12 with about 160 migrants – others have joined the caravan during the journey
- Where: most are a thousand miles away from the US at this writing, although a splinter group of mostly LGBTQ refugees has reached the border and is preparing to ask for asylum.
- Why: to escape poverty, discrimination, persecution, and gang violence
- How: many are walking, and some are getting rides in slow-moving trucks

You could read all these facts and still have concerns about the migrant caravan. Responsible and well-meaning adults can have different opinions on what the situation means and what to do about it, but putting in the time to consider the details improves the quality of the picture, increasing the probability that the discussion will be based on facts and not on fear, anger, or personal mythology.

The high velocity and low overhead of the Internet means that no one is really watching to ensure that these important details, which are the backbone of professional journalism, are addressed before the post or tweet goes live, so it's up to you to be the editor of everything you read online. You need to ride with the five Ws and the six honest serving men whenever you log in.

But the vague, emotionally manipulative stories are only part of the problem. Also cluttering the landscape is a new kind of news story, pioneered by Russian trolls and enabled by morally absent western politicians, that offers up an official-looking presentation of this who-what-when-where-why information, but with so-called "facts" that are simply false (what we used to call *lies*). Some of these stories might actually look like real news stories, which makes them even more dangerous. In this case, you need to reach for another tool in the old-school journalism toolkit: look for additional, independent, corroborating sources.

Either way, keep the serving men close at hand: The Internet doesn't really eliminate the need for editorial oversight: it just offloads the responsibility from the publisher to you the reader.



Joe Casad,
Editor in Chief



WHAT'S INSIDE

A **free mapping service** is a wonderful thing, but you might not find all the information you need in OpenStreetMap's standard view. A number of other free mapping projects extend and enhance OpenStreetMap data to add new features and services. This month, we tour some useful mapping tools based on OpenStreetMap.

Other highlights include:

- **Kernel Disposal** – Nuke those old kernels taking up space on your hard drive (page 32).
- **Out of Memory Killer** – This important component keeps the kernel from running out of memory – but is there a better way? (page 46)

Check out MakerSpace for a lesson in wearables, and read on to LinuxVoice for a report on the dark market for security exploits.

SERVICE

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

NEWS

08 News

- IBM Acquires Red Hat
- Fedora 29 and Ubuntu 18.10 Released
- Redis Labs Modules Forked
- Microsoft Offers Its Patent Portfolio

12 Kernel News

- Linus Returns to the Kernel
- Coscheduling and Intel Vulnerabilities

16 2018 OS Summit

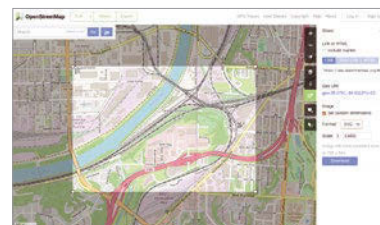
The European leg of the Open Source Summit took place at the Edinburgh International Conference Centre on October 22-24, 2018. The conference was co-located with the Embedded Linux Conference and the OpenIoT Summit.



COVER STORIES

18 OSM-Based Online Services

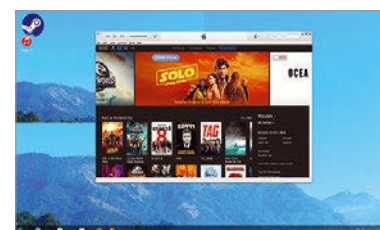
A variety of online services rely on the data collected by the OpenStreetMap project. This article introduces you to the most useful options.



REVIEWS

28 Zorin OS

If you are looking to migrate from Windows and Mac OS to Linux, but you still need Windows compatibility and applications, Zorin OS might be the solution.



IN-DEPTH

32 Getting Rid of Old Kernels

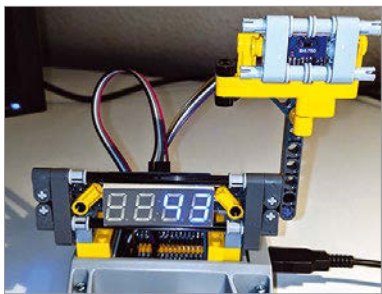
When you update the kernel, the old version remains on the disk. If you clean up, the reward could be several hundred megabytes of freed space.

36 Command Line – CSS

Cascading Style Sheets can provide a sophisticated layout for articles and books. A few tips will help you get started.

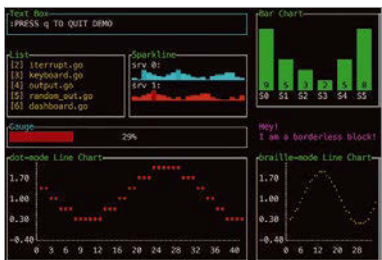
39 Charly's Column – Hue and Rasp Pi

Since his Trådfri Smarthome article over a year ago, sys admin columnist Charly has been receiving messages from readers with two questions: "Can you do that with the Philips' Hue system?" and "Can this also be done with a normal brightness sensor?" Yes and yes!



42 Programming Snapshot – termui

Even command-line lovers appreciate a classic terminal UI. Mike Schilli shows how to whip up a Go program that dynamically displays network interfaces and IPs.



46 OOM Killer

When a Linux system runs out of memory, a special agent called the out-of-memory killer rushes to its aid.

52 WebAuthn

FIDO2 authentication with WebAuthn may be sounding the end of the password age.

MAKERSPACE

58 GoPiGo3

The GoPiGo3 kit provides components and software for a small robot car with a Raspberry Pi brain.



62 Open Hardware – Palitra

This special portable keyboard is designed for entering keyboard shortcuts.

66 Wearable Tech

Create some interesting wearable projects with an Android device, an armband case, headphones, and MIT's free App Inventor package.



TWO TERRIFIC DISTROS
DOUBLE-SIDED DVD!

SEE PAGE 6 FOR DETAILS

LINUXVOICE

69 Welcome

This month in Linux Voice.

71 Doghouse – IoT

In conjunction with Caninos Loucos, maddog helps develop a hardware platform for the Brazilian IoT project.

72 Retrosare

Communication on the Internet is not as secure as you might think. We show you how to eliminate curious eavesdroppers.



76 Military-Malware Complex

Modern cyberwarfare and its resulting monetary allocations have significantly impacted the exploit market.

84 FOSSPicks

This month Graham looks at draw.io, musikcube, Whalebird, and more.



92 Tutorial – Plasma

Get started with KDE's powerful desktop.

On the DVD



fedora

29

Ubuntu 18.10
Desktop (64-bit)

ISSUE 218 JAN 2019
LINUX
MAGAZINE

DVD
ROM

TWO TERRIFIC
DISTROS

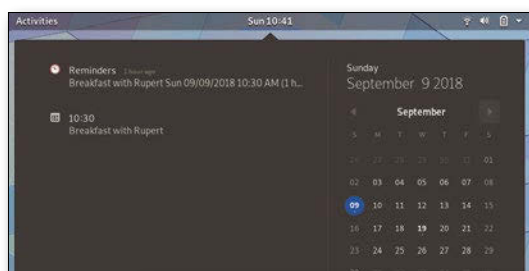
DOUBLE-SIDED
DVD!

Ubuntu 18.10 "Cosmic Cuttlefish" Desktop 64-bit

Ubuntu Linux is a popular Debian-based distro with a massive user base and active community support. Ubuntu is a versatile system that includes codecs and drivers for smooth out-of-the-box installation on a great variety of systems. The latest release comes with Linux kernel 4.18, Gnome 3.30, and OpenSSL 1.1.1 with support for TLS 1.3. Other highlights include smoother desktop zoom, better support for VeraCrypt disk encryption, and updates to the renamed Yaru community theme, which is now the default.

Fedora 29 Workstation 64-bit

The Red-Hat-sponsored Fedora is an all-free Linux that targets software developers and end users. Many new features that will one day appear in the enterprise-grade RHEL series get their start in Fedora. The latest release features improved desktop performance, better screen-sharing control, and automatic updates for installed flatpaks. Other enhancements include better VeraCrypt support and a new podcast management app. The Boxes virtual machine manager can now connect to remote Windows sessions.



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

Additional Resources

- [1] Ubuntu 18.10 Release Notes: <https://wiki.ubuntu.com/CosmicCuttlefish/ReleaseNotes>
- [2] Ubuntu Desktop Guide: <https://help.ubuntu.com/stable/ubuntu-help/index.html>
- [3] Ubuntu Forums: <https://ubuntuforums.org/>
- [4] Fedora 29 Release Notes: <https://docs.fedoraproject.org/en-US/fedora/f29/release-notes/>
- [5] Ask Fedora: <https://ask.fedoraproject.org/en/questions/>
- [6] Fedora Forum: <https://fedoraforum.org/>

Conference and Summit 2019



Open Source Software + Embedded Systems + Cloud + Containers



openSUSE Conference 2019

Nürnberg, Germany

May 24 - 26, 2019

openSUSE Summit at SCaLE17x

California, United States

March 08 2019



openSUSE Summit Nashville

Nashville, United States

April 05 - 06, 2019



NEWS

Updates on technologies, trends, and tools

THIS MONTH'S NEWS

- 08 • IBM Acquires Red Hat
- 09 • Fedora 29 and Ubuntu 18.10 Released
 - Redis Labs Modules Forked
 - More Online
- 10 • Microsoft Offers Its Patent Portfolio

IBM Acquires Red Hat

In a surprise move, IBM has announced that it is acquiring Red Hat for \$34 billion. A primary reason for the purchase is to support IBM's hybrid cloud and Kubernetes business. IBM is a distant fourth player in the public cloud space, behind AWS, Azure, and Google Compute Engine.

Red Hat's primary bread winner is still Linux, but the Linux business has kind of reached a saturation point. The real year-on-year growth is coming from the emerging business, which is the cloud.

Red Hat doesn't have any public cloud offerings; its strength is Kubernetes-based OpenShift, which enables customers to embark on their hybrid cloud journey. However, Red Hat can grow only so much as customers move to public cloud configurations. Red Hat can't scale to address the growing market, and it is limited in how much it can invest in other hot emerging technologies like Machine Learning and IoT.

"We can only do so much," Red Hat President and CEO Jim Whitehurst told me in a previous interview.

"Joining forces with IBM will provide us with a greater level of scale, resources, and capabilities to accelerate the impact of open source as the basis for digital transformation and bring Red Hat to an even wider audience – all while preserving our unique culture and unwavering commitment to open source innovation," said Whitehurst.

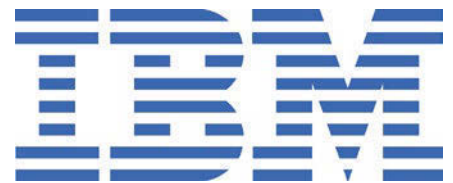
IBM said that joining with Red Hat will help them help clients create cloud-native business applications faster, drive greater portability and security of data and applications across multiple public and private clouds, and provide consistent cloud management.

To achieve this, the two companies will leverage key open source technologies such as Linux, containers, Kubernetes, multi-cloud



management, cloud management, and automation.

"The acquisition of Red Hat is a game-changer. It changes everything about the cloud market," said Ginni Rometty, IBM chairman, president, and chief executive officer. "IBM



will become the world's #1 hybrid cloud provider, offering companies the only open cloud solution that will unlock the full value of the cloud for their businesses.”

Although the cloud is at the heart of this acquisition, the change won't affect the work Red Hat does in the Linux and open source space. In fact, IBM itself is among the top 10 contributors to the Linux kernel. During a conference call, IBM and Red Hat told me that there won't be any impact on open source projects maintained by Red Hat, including Gnome, Fedora, CentOS, and others.

Red Hat will continue to run as an independent entity under the leadership of Whitehurst, within the Hybrid Cloud business of IBM.

Fedora 29 and Ubuntu 18.10 Released

October is the time of the year when users get to play with new versions of Ubuntu and Fedora. Canonical announced Ubuntu 18.10, and the Fedora community announced Fedora 29. Both are Gnome-based distributions. Ubuntu has focused on faster boot times and improved support for new hardware; Fedora has focused on improving its modular design.



“Modularity helps make it easier to include alternative versions of software and updates than those shipped with the default release, designed to enable some users to use tried-and-true versions of software while enabling others to work with just-released innovation without impacting the overall stability of the Fedora operating system,” according to a Fedora press release.

Fedora comes in three editions: Workstation, Cloud, and Atomic Host. The latest version of Fedora's desktop-focused edition provides new tools and features for general

users, as well as developers, with the inclusion of Gnome 3.30. Fedora is putting its weight behind Flatpack.

Ubuntu also comes in different editions: Ubuntu Desktop, Ubuntu Server, Ubuntu Cloud, and Ubuntu for IoT. There are different flavors of Ubuntu that support various desktop environments, including KDE Plasma, LXQt, etc.

Snap is the default app packaging and delivery mechanism of Ubuntu that competes with Flatpack. Canonical said that Ubuntu's Linux app store includes 4,100 snaps from over 1,700 developers with support across 24

Linux distributions. 18.10 enables native desktop controls to access files via the host system.

While Fedora remains a distribution for developers (Linus Torvalds himself uses Fedora), Ubuntu still appeals to a wider audience, from gamers to enterprise customers.



Redis Labs Modules Forked

As expected, developers from the desktop projects Fedora and Debian have forked the modules that database vendor Redis Labs put under the Commons Clause.

The Commons Clause is an extra license rider that prohibits the user from "selling" the software, with "selling" defined as including selling services such as hosting and consulting. According to Redis Labs and the creators of the Commons Clause, the rider was created to prevent huge hosting companies like Amazon from using the code without contributing to the project. Unfortunately, the license also has the effect of making the Redis Labs modules incompatible with the open source licenses used with Linux and other FOSS projects.

To fix the problem, Debian and Fedora came together to fork these modules. Nathan Scott, Principal Software Engineer at Red Hat, wrote on a Google Group, "...we have begun collaborating on a set of module repositories forked from prior to the license

MORE ONLINE

Linux Magazine

www.linux-magazine.com

Paw Prints • Jon “maddog” Hall

IBM Purchase of Red Hat Software: There is No Fear Except Fear Itself – with Thanks to FDR

IBM bought Red Hat Software. The world wide web is alive with the news, and many of the people who have worked and used Red Hat in the last 25 years are lamenting the “fall” of their beloved company and software.

ADMIN HPC

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

Slurm Job Scheduling System • Jeff Layton

In previous articles, I examined some fundamental tools for HPC systems, including pdsh (parallel shells), Lmod environment modules, and shared storage with NFS and SSHFS. One remaining, virtually indispensable tool is a job scheduler.

ADMIN Online

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

TLS 1.3 and the Return of Common Sense

Filipe Pereira Martins and Anna Kobylinska

After a decade in service, TLS 1.2 is showing many signs of aging. Its immediate successor, TLS 1.3, has earned the approval of the IETF. Some major changes are on the way.

Make Better Use of Prometheus with Grafana, Telegraf, and Alerta • Martin Loschwitz

The Prometheus monitoring tool might not always look like one of the Titans, but add-ons like Alerta or Telegraf can improve its looks.

Exploring the Most Famous Performance Tool Federico Lucifredi

Take your performance tuning into the cloud with the top utility.

change. We will maintain changes to these modules under their original open source licenses, applying only free and open fixes and updates.”

It was an expected move. When license changes are made to any open source project, often some open source community jumps in and forks the project to keep a version fully compatible with the earlier open source license. The fork means commercial vendors like Amazon will still be able to use these modules without contributing anything to Redis Labs or the newly forked project. However, not all forks are successful. It's not the license that matters. What matters is the expertise of the developers who write and maintain the codebase. Google once forked Linux for Android, but eventually ended up merging with the mainline kernel.

In a previous interview, Redis Labs told me that they were not sure whether adding the Commons Clause to these licenses would work or not; they already tried the Affero GPL (AGPL) license, which is also designed to address the so-called application

service provider loophole that allows cloud vendors to avoid contributing back their changes, but the move to the AGPL didn't help them get vendors like Amazon to contribute.

Redis Labs added the Commons Clause to only those modules that their staff wrote; there is no change to the modules written by external parties.

© Dmitry Sunagatov, Fotolia.com



Microsoft Offers Its Patent Portfolio

In a surprise and historical move, Microsoft has released its entire patent portfolio to Open Innovation Network (OIN) by joining the organization. Microsoft has released all 60,000 patents to OIN.

“We bring a valuable and deep portfolio of over 60,000 issued patents to OIN for the benefit of Linux and other open source technologies,” said Erich Andersen, corporate vice president, deputy general counsel.

These patents also include those 235 patents that Microsoft once claimed were infringed upon by the Linux kernel. Linus Torvalds had dismissed those claims stating, “Microsoft just made up the number.”

It's a major U-turn for Microsoft, which has a history of exploiting patents as a weapon against Linux players. This move brings an end to the long hostility between Linux and Microsoft.

There are more than 2,650 members, including numerous Fortune 500 enterprises, that make OIN the largest patent non-aggression community.

OIN has created a massive pool of patents affecting Linux and open source projects. The organization offers these patents on a royalty-free basis to member organizations. Companies not yet members of OIN can also tap into its pool of patents if they promise not to assert its patents against the Linux system.

Back in 2005, OIN was created by a group of companies with vested interests in open source.

The goal was to fend off any patent attacks on open source companies. Founding members included IBM, NEC, SUSE/Novell, Philips, Red Hat, and Sony.

Microsoft has around 90,000 patents, but over 30,000 as still pending with the US Patent Office. Once those patents are approved, they will also become part of the OIN pool.



Microsoft



What?!
I can get my
issues
SOONER?



Available anywhere, anytime!

Sign up for a digital subscription and enjoy the latest articles on trending topics, reviews, cool projects and more...

shop.linuxnewmedia.com/digisub

Zack's Kernel News



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

By Zack Brown

Linus Returns to the Kernel

Linus Torvalds's self-imposed exile from kernel development lasted exactly one week, ending on October 23 when he began to catch up on kernel merges. He had originally decided to take a break to think about his sometimes harsh treatment of Linux developers.

Given the extreme reactions to his departure – or at least to the accompanying “code of conduct” that entered the kernel source tree at the same moment – his return was greeted peacefully. Mostly he addressed the question

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**.

of whether developers would prefer to receive an email acknowledgement when they submitted merge requests; the responses ranged all over the place, with some saying yes, some saying no, some explaining their own recipes for handling merge requests, and some suggesting various techniques and alternatives for automating various parts of Linus's workflow.

The real debate will come when, for example, someone disingenuously tries to get “security” code into the kernel that could be used to lock users out of controlling their own systems. How will Linus respond when someone refuses to acknowledge the validity of technical objections and continues to push for their particular patch?

My personal guess is that Linus and other top developers may now start to freeze certain people out. Instead of posting harsh responses, they may simply fail to reply to those posts at all. They won't make any public argument against certain approaches to the Linux kernel; those approaches will be granted a de facto legitimacy, even though their code won't immediately go into the kernel tree.

Coscheduling and Intel Vulnerabilities

Jan H. Schön herr wanted to extend the Completely Fair Scheduler (CFS) to support coscheduling, which is a technique of scheduling related processes on separate CPUs. It's not necessarily better than other scheduling approaches, but it has advantages and disadvantages depending on the situation and on the specific implementation details of each different scheduling approach.

Jan gave several examples of possible use cases that would benefit from coscheduling. For example, “if we can derive subsets of tasks, where tasks in a subset don't interfere much with each

other when executed in parallel, then co-scheduling can be used to realize this more efficient schedule. And ‘resource’ is a really loose term here: from execution units in an SMT system, over cache pressure, over memory bandwidth, to a processor’s power budget and resulting frequency selection.”

Jan was not trying to implement co-scheduling for all circumstances. In particular, Jan’s code would coschedule tasks that were not already being handled by other scheduling techniques in CFS; coscheduling would not be the primary scheduling approach.

As another caveat, Jan explained, “The collective context switch from one coscheduled set of tasks to another – while fast – is not atomic. If a use case needs the absolute guarantee that all tasks of the previous set have stopped executing before any task of the next set starts executing, an additional handshake/barrier needs to be added.”

Nishanth Aravamudan did some testing and found that he could hang the system under some circumstances. Jan looked at Nishanth’s setup and felt that it should work fine – though he acknowledged that he too saw a system lockup with the same configuration. The two went back and forth hunting the bug, and Jan eventually solved several issues that had contributed to the lockups.

In his initial post, Jan had said that Peter Zijlstra had once called coscheduling a “scalability nightmare waiting to happen.” Meanwhile, Peter also replied to Jan’s initial post, saying, “this isn’t anywhere near something to consider merging.”

In particular, Peter said that there were certain issues in the existing kernel cgroup code, used for virtual machines (VMs), that made it hard to scale upward to many VMs – and he said Jan’s coscheduling code made these particular scalability issues “a ton worse” and “many times worse.” He felt that the cgroup code needed to be thoroughly gone through, cleaned up, and optimized, before anything like Jan’s current patch could even be considered.

Peter added some context, saying, “I detest cgroups; for their inherent complexity and the performance costs associated with them. ... It is after all, per-

fectly possible to run a KVM thingy without cgroups.”

He felt that CFS itself was entirely the wrong place for the kind of “gang scheduling” Jan had in mind. He said such things were possible, “but not within the confines of something like CFS; they are also fairly inefficient because, as you do note, you will have to explicitly schedule idle time for idle vCPUs.”

Peter was also unsatisfied with Jan’s technical explanation of the coscheduling implementation details, saying “You gloss over a ton of details here; many of which are non trivial.” He said, “Unless you have solid suggestions on how to deal with all of them, this is a complete non-starter.”

He went into a bunch of technical details here, regarding specific areas of Jan’s patch that Peter felt had no clear design. Peter also had a negative critique of the patch code itself, saying: “What about that atrocious locking you sprinkle all over the place? ‘Some additional lock contention’ doesn’t even begin to describe that horror show. Hint: We’re not going to increase the lockdep subclasses, and most certainly not for scheduler locking.”

Jan replied that, “This patch set should ‘just’ give the user the additional ability to coordinate scheduling decisions across multiple CPUs. At least, that’s my goal. If someone doesn’t need it, they don’t have to use it. Just like task groups.”

On the other hand, Jan continued, if someone did want to use coscheduling, the patch would present them with the ability to experiment with coordinated scheduling decisions. He pointed out that there was a lot of interesting research about the kind of benefits that coscheduling could provide if it were ever implemented – so why not implement it? Jan reiterated that “existing scheduler features, like preemption, (should) still work as before with this variant of coscheduling, with the same look and feel.”

Jan acknowledged that – as Peter had said – the patch wasn’t ready to be merged into the kernel tree. He wasn’t submitting it for inclusion, but was just trying to start a discussion about the mechanics and implementation.

Jan also addressed a number of Peter’s technical objections – also reiterating that many of these were made against

code that was acknowledged to be not yet ready for inclusion.

Peter replied again with a bit more of his own context. He explained, “I have, of course, been looking at (SMT) co-scheduling, specifically in the context of L1TF, myself. I came up with a vastly different approach. ... Note, that even though I wrote much of that code, I don’t particularly like it either.”

L1TF is a security hole in Intel chips, revealed in August 2018, which comes on the heels of a number of other very serious hardware flaws that have been discovered relatively recently in Intel chips. In the case of L1TF, the vulnerability allows hostile code to make an educated guess about what the CPU is likely to do next. It’s specifically a vulnerability related to process scheduling.

One of Peter’s original objections had been that Jan’s sole motivation for writing this patch was to try to address L1TF, which Peter felt could really be addressed in better ways. Although later, Peter relented and acknowledged that regarding Jan’s motivations, “I might have jumped to conclusions here.”

Still, after Peter’s acknowledgement, Rik van Riel wanted to know what other motivations beyond L1TF Jan had for producing the patch. Specifically, “What are the other use cases, and what kind of performance numbers do you have to show examples of workloads where coscheduling provides a performance benefit?”

Jan replied at length (complete with bibliographic references):

Many coscheduling use cases are not primarily about performance.

Sure, there are the resource contention use cases, which are barely about anything else. See, e.g., [1] for a survey with further pointers to the potential performance gains. Realizing those use cases would require either a userspace component driving this, or another kernel component performing a function similar to the current auto-grouping with some more complexity depending on the desired level of sophistication. This extra component is out of my scope. But I see a coscheduler like this as an enabler for practical applications of these kind of use cases.

If you use coscheduling as part of a solution that closes a side channel, perfor-

mance is a secondary aspect, and hopefully we don't lose much of it.

Then, there's the large fraction of use cases, where coscheduling is primarily about design flexibility, because it enables different (old and new) application designs, which usually cannot be executed in an efficient manner without coscheduling. For these use cases performance is important, but there is also a trade-off against development costs of alternative solutions to consider. These are also the use cases where we can do measurements today, i.e., without some yet-to-be-written extra component. For example, with coscheduling it is possible to use active waiting instead of passive waiting/spin-blocking on non-dedicated systems, because lock holder preemption is not an issue anymore. It also allows using applications that were developed for dedicated scenarios in non-dedicated settings without loss in performance – like an (unmodified) operating system within a VM, or HPC code. Another example is cache optimization of parallel algorithms, where you don't have to resort to cache-oblivious algorithms for efficiency, but where you can stay with manually tuned or auto-tuned algorithms, even on non-dedicated systems. (You're even able to do the tuning itself on a system that has other load.)

Now, you asked about performance numbers, that I have.

If a workload has issues with lock-holder preemption, I've seen up to 5x to 20x improvement with coscheduling. (This includes parallel programs [2] and VMs with unmodified guests without PLE [3].) That is of course highly dependent on the workload. I currently don't have any numbers comparing coscheduling to other solutions used to reduce/avoid lock holder preemption, that don't mix in any other aspect like resource contention. These would have to be micro-benchmarked.

If you're happy to compare across some more moving variables, then more or less blind coscheduling of parallel applications with some automatic workload-driven (but application-agnostic) width adjustment of coscheduled sets yielded an overall performance benefit between roughly 10% to 20% compared to approaches with passive waiting [2]. It was roughly on par with pure space-partitioning ap-

proaches (slight minus on performance, slight plus on flexibility/fairness).

I never went much into the resource contention use cases myself. Though, I did use coscheduling to extend the concept of "nice" to sockets by putting all niced programs into a coscheduled task group with appropriately reduced shares. This way, niced programs don't just get any and all idle CPU capacity – taking away parts of the energy budget of more important tasks all the time – which leads to important tasks running at turbo frequencies more often. Depending on the parallelism of niced workload and the parallelism of normal workload, this translates to a performance improvement of the normal workload that corresponds roughly to the increase in frequency (for CPU-bound tasks) [4]. Depending on the processor, that can be anything from just a few percent to about a factor of 2.

Rik replied, "I like the idea of having some coscheduling functionality in Linux, but I absolutely abhor the idea of making CFS even more complicated than it already is. The current code is already incredibly hard to debug or improve. Are you getting much out of CFS with your current code? It appears that your patches are fighting CFS as much as they are leveraging it."

He felt that coscheduling would probably be much better as its own scheduler class and leave it out of CFS entirely. In another email, he reiterated, "CFS is already complicated enough that it borders on unmaintainable. I would really prefer to have the coscheduler code separate from CFS, unless there is a really compelling reason to do otherwise."

Jan replied that he still felt that the coscheduling code was not so terrible in CFS, and that really it just leveraged all the hard work that had already gone into CFS. But, he went on, if he were going to consider taking the coscheduling code out of CFS, "I'd overhaul the scheduling class concept as it exists today. Instead, I'd probably attempt to schedule instantiations of scheduling classes. In its easiest setup, nothing would change: one CFS instance, one RT instance, one DL instance, strictly ordered by priority (on each CPU). The coscheduler as it is posted (and task groups in general) are effectively some form of multiple CFS instances being governed by a CFS in-

stance. This approach would allow, for example, multiple CFS instances that are scheduled with explicit priorities; or some tasks that are scheduled with a custom scheduling class, while the whole group of tasks competes for time with other tasks via CFS rules."

There was quite a bit of implementation discussion surrounding the debate. The primary debate ended roughly here though, and it seems clear that there is plenty of support for coscheduling in general. Efforts to route around LITF are definitely motivating factors, although coscheduling has its own independent appeal as well.

The big issue mostly seems to center around how to implement coscheduling in a way that is not psychotically complex. Apparently, all the code it touches or might touch is already bordering on insane. For the people objecting to Jan's patch – primarily Rik and Peter – their main concern seems to be just finding a way to maintain the code once it goes into the kernel. ■■■

Info

- [1] Zhuravlev, S., J. C. Saez, S. Blagodurov, A. Fedorova, and M. Prieto. Survey of scheduling techniques for addressing shared resources in multi-core processors. *ACM Computing Surveys*, 2012;45(1):4.1-4.28
- [2] Schönherr, J. H., B. Juurlink, and J. Richling. TACO: A scheduling scheme for parallel applications on multicore architectures. *Scientific Programming*, 2014;22(3):223-237
- [3] Schönherr, J. H., B. Lutz, and J. Richling. "Non-intrusive coscheduling for general purpose operating systems." *In: Proceedings of the International Conference on Multicore Software Engineering, Performance, and Tools (MSEPT '12)*, ser. Lecture Notes in Computer Science, vol. 7303 (2012). Berlin/Heidelberg, Germany: Springer, pp. 66-77
- [4] Schönherr, J. H., J. Richling, M. Werner, and G. Mühl. "A scheduling approach for efficient utilization of hardware-driven frequency scaling." *In: M. Beigl and F. J. Cazorla-Almeida, eds. Workshop Proceedings of the 23rd International Conference on Architecture of Computing Systems (ARCS Workshops)*, 2010, Berlin, Germany: VDE Verlag, pp. 367-376

COMPLETE YOUR LIBRARY

Order a digital archive bundle and save at least **50% off** the digisub rate!



ORDER YOURS TODAY!
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!

Open Source Summit – Edinburgh 2018

OPEN SOURCE EVERYWHERE

The European leg of the Open Source Summit took place at the Edinburgh International Conference Centre on October 22-24, 2018. The conference was co-located with the Embedded Linux Conference and the OpenIoT Summit. *By Richard Ibbotson*

The the Open Source (OS) Summit [1] kicked off when Jim Zemlin, Executive Director of the Linux Foundation, gave a general introduction to the state of open source software. Throughout the conference, Jim made several more appearances, with helpful updates on what was happening each day.

Keynotes

On the first day, Shuli Goodman delivered her keynote, “Wanted, 10,000 Developers to Electrify the Planet.” She is Executive Director of LF Energy, a Linux Foundation open source coalition to “... accelerate and transform the world’s relationship to energy” and make the planet more energy efficient. For more information about this talk

and others, have a look at the Linux Foundation slides page [2].

The keynote speech by Arjan van de Ven, Intel Fellow and Director of Linux Systems Engineering, was about “Software-Defined Everything.” He stated that in a software-defined world, with a feature added here and a feature added there, the development model breaks down, requiring a complete rebuild of the software stack. Because any use case in the cloud begins with Linux, he used his “passion project,” Clear

Linux, as the Linux distro at the bottom of this new stack and Acorn, a functionally safe and capable hypervisor, as the isolation layer of the new vertical stack integration.

Jonathan Corbett, Executive Editor for LWN.net and a kernel developer, presented the kernel report, in which he gave a brief summary of recent kernel releases. He noted that more than 13,000 changes have taken place in the kernel in the past 12 months and that breakdowns in the 4.15 development cycle were caused by Meltdown and Spectre hardware vulnerabilities. Unlike the Meltdown solution, which had patches in the pipeline when it was disclosed, Corbett explained that the Spectre fixes were completed in secrecy and in isolation instead of in the usual way, in which a team of knowledgeable people are gathered together and given the information they need to effect a solution. The results were different solutions by multiple distributors, none of which survived in the mainline kernel, and developer burnout and frustration. The hope is that lessons have been learned.

One Tuesday morning keynote speaker was Eric Berlow, an ecologist, network scientist, and Cofounder and Chief Science Officer of Vibrant Data. Berlow is internationally recognized for his research on ecological complexity, and his



Jim Zemlin



Shuli Goodman



Arjan van de Ven



Eric Berlow



might think of in the world of open source software. One of the more interesting presentations by Andrea Grandi, “Lessons Learned Open Sourcing the UK Government,” was about the challenges and problems of moving the UK Government Digital Service to open coding.

Ben Hall took to the stage to talk about Katacoda container security. His talk, “Three Years of Lessons from Running Potentially Malicious Code Inside Containers” was an interesting account of his involvement in the world of DevOps and containers and how the Live environments for cloud-native technologies are vulnerable through the

Linux build environment, and Isar, an image generator for embedded systems, which both use Deby, a Linux distribution for embedded systems.

The rest of the conference was packed solid with fascinating talks, such as “Coccinelle: 10 Years of Automated Evolution and Bug Finding in the Linux Kernel” with Julia Lawall of Inria, “10 Years of the Industrial I/O Kernel Subsystem” with Jonathan Cameron from Huawei, and “Establishing Image Provenance and Security in Kubernetes” by Adrian Mouat of Container Solutions, among many others too numerous to list here. As usual, participants got to attend many social events that always take place at OS Summit. ■■■

Info

- [1] Schedule: <https://events.linuxfoundation.org/events/open-source-summit-europe-2018/program/schedule/>
- [2] Slides: <https://events.linuxfoundation.org/events/open-source-summit-europe-2018/program/slides/>
- [3] YouTube videos: <https://www.youtube.com/playlist?list=PLbzoR-pLrL6qThA7SAbhVfuMbjZsJX1CY>



two TED talks on finding hidden patterns in complex data have received more than 2 million views. In his keynote speech, “The Future of AI is Data ... In More Ways than You Think,” he said “It’s not about the algorithms; it’s actually all about the data.” While acknowledging such problems as biased facial recognition software and the non-transparency of AI algorithms, which he notes were pointed out in the book *Weapons of Math Destruction* by Cathy O’Neil, he argued that “healthy AI depends on a healthy data ecosystem.” He then gave some examples of how the data behind complex algorithms give unexpected, even biased results. You can see his talk and others on the OS Summit YouTube channel [3].

On Wednesday morning Alexander Nitz, a researcher at the Max Planck Institute for Gravitational Physics, presented “Astronomy with Gravitational Waves,” an excellent example of open science with open source software.

Presentations

In addition to other keynotes, the conference was filled with high-quality presentations about just about anything you

web browser, with the possibility of malicious code being executed from inside the container. He addressed how to monitor and secure these systems and how to prepare for attacks.

The “Debian & Yocto: State of the Art” presentation by Kazuhiro Hayashi, Toshiba Corporation; Manuel Traut, Linutronix GmbH; and Baurzhan Ismagulov, ilbers GmbH, addressed how the Debian distribution is used in the industrial environment for building GNU/Linux-based products, joining together the benefits of existing distributions and tools. Specifically, they spoke about a collaboration in building the Debian-based products ELBE, an embedded





Online services that extend the capabilities of OpenStreetMap

Useful

A variety of online services rely on the data collected by the OpenStreetMap project. This article introduces you to the most useful options. *By Karsten Günther*

OpenStreetMap (OSM) [1] is one of the largest and most popular community web projects. In contrast to a tool like Wikipedia, however, OSM users rarely see the project's website. Instead, geo-information systems such as route planners, navigation apps and devices, or outdoor trackers integrate OSM's data as part of their own presentation.

If you visit the OSM website directly, you can click on the *Layers* button on the right side to display alternative map forms, such as Cycle Map, Transport Map, or Humanitarian Map (Figure 1). In addition, you can display current information about the maps by checking the *Map Notes* checkbox.

The contents are shown as soon as you mouse over or click on the little flag in the map. Many layers come with legends that you can enable by clicking on the button with the *i* to the right of the map.

Unlike commercial map services like Google Maps or Bing Maps, OSM lets you export your map data. Be sure to comply with copyright and license requirements [2]. Some forms of data are available using the *Export* button in the top left corner of the OSM main view. If you want to save the map image, click on the *Share* button in the sidebar (Figure 2). OSM supports several image formats, including PNG or JPG, but also artifact-free scalable vector graphics in SVG and PDF formats.

SVG is particularly well-suited for further processing, typically with supporting applications such as Inkscape. If you prefer Gimp, make sure you enable the *Import paths* option



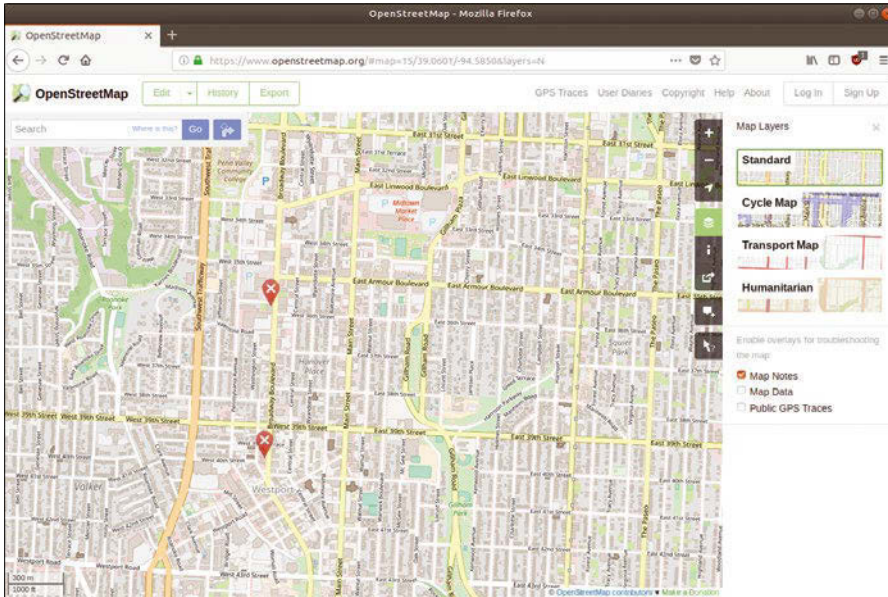


Figure 1: OSM offers a number of alternative map forms. Map Notes shows additional information on the map.

when importing. *Import paths* gives you direct access to all the fonts, lines, and other elements of the map.

The easy availability of OSM data means other projects can build their own mapping services around the large and comprehensive OSM dataset. A community group could create a project that focuses on a specific feature or activity or provide value-added data for a specific region. In the true spirit of open source, some of these services tie back in with OSM and are available directly from the OSM interface in addition to having their own homepages.

This article tours some services that make use of OSM data. I'll also highlight some useful options within the OSM user interface that provide additional information for users with specific needs.

OpenCycleMap

OpenCycleMap [3] is a map tool for cyclists. You can access OpenCycleMap through OSM or via the project's own homepage. OpenCycleMap is drawn sparingly and commented even more sparingly. However, the map highlights cycle paths and identifies official long-distance paths (Figure 3).

Like OSM, OpenCycleMap is continuously updated. Route changes such as construction detours are often reflected

in the map within only a few days. The map is also suitable for planning some longer tours. OpenCycleMap identifies points of interest such as shelters and sheds, water points and wells, bicycle shops, and some cafés and restaurants.

Waymarked Trails

If you are planning longer cycle tour in Europe or a tour on long-distance cycle route, such as the North Sea Cycle Route (EuroVelo 12) [4], it is easier to find your way around with the help of specialized sites such as Waymarked Trails [5] and other long-distance alternatives [6]. Waymarked Trails offers a path through well-known and sign-posted routes for hiking, cycling, MTB, ski touring, or skating (Figure 4).

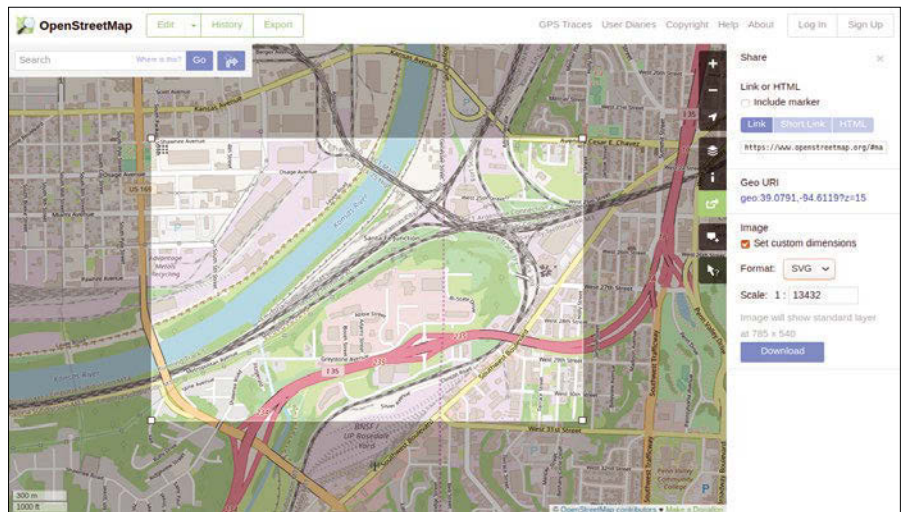


Figure 2: You can export an OSM image in both bitmap (PNG, JPG) and vector graphic (PDF, SVG) formats.

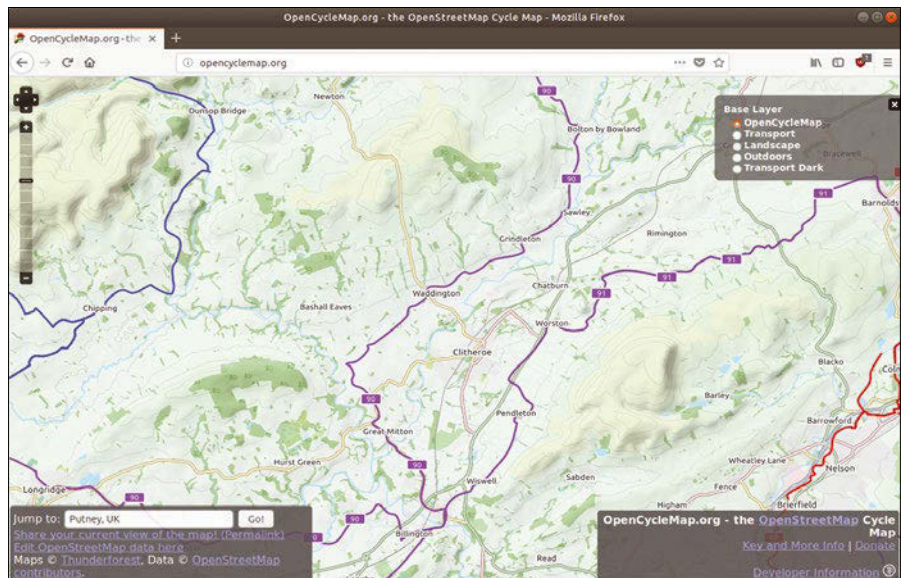


Figure 3: OpenCycleMap has different layers (top right). The map also contains practical points of interest, such as drinking water sources or bicycle shops.

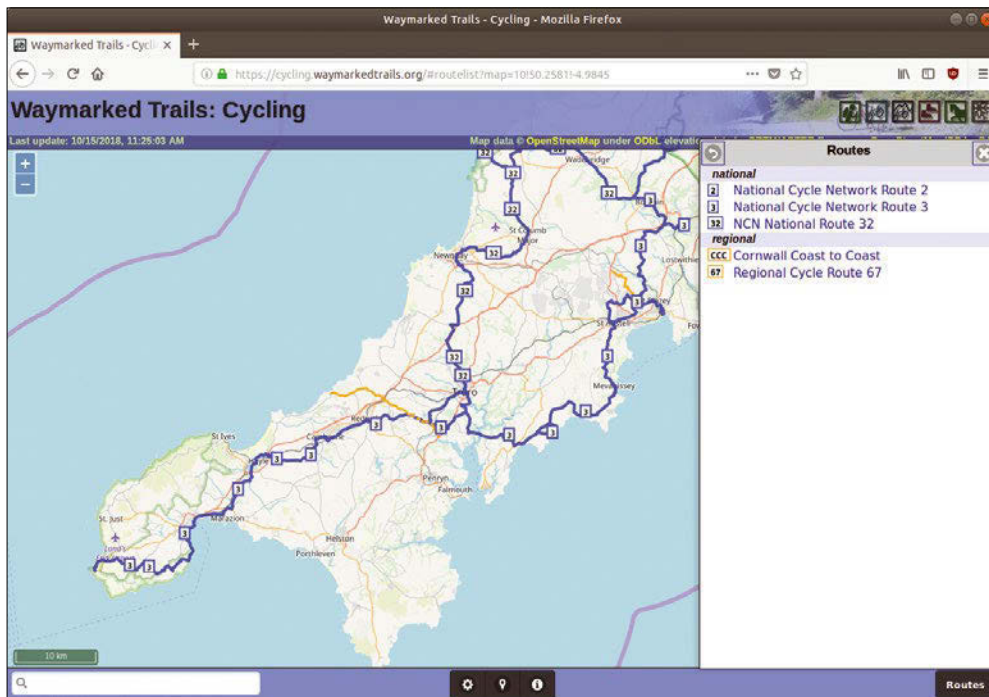


Figure 4: Waymarked Trails focuses on European long-distance cycle routes. The *Routes* button lists the routes displayed in the current map section.

The actual use of Waymarked Trails is only revealed when you zoom into an area and use the *Routes* button (bottom right) to find the marked routes available in the displayed area. In the detail view, the page also offers a download option in the form of a GPX track for most routes.

Although its European maps are far more comprehensive, Waymarked Trails does include some major routes in North America.

Naviki

For a long time, Naviki [7] was regarded as an “insider tip” among route planners for bicycles, since the project was developed at Münster University of Applied Sciences and did not pursue any commercial interests. In the meantime, however, the site has been further developed and is hosted by Beemo GmbH, a spin-off of the Laboratory for Software Engineering at Münster University of Applied Sciences.

Naviki’s charm lies in the simple operation of the software, especially on computers with a large display. It is very easy to create a cycle route: First, right-click on the map to mark the start and destination. Then select the desired vehicle type (Figure 5). *Everyday* creates medium-length routes that try to avoid major roads. *Shorter Route* does the opposite: In order to minimize the

length, altitude, and road type (single trail, cycle path, or country road). Naviki does not go further into the details of the routing algorithm. It is therefore a good idea to first get a feeling for how it works by creating routes in familiar territory.

Automatic route planning with Naviki produces useful to good route suggestions, but for the “perfect” bike tour, you usually have to adjust the route a little by hand. Grab the route with the mouse pointer and drag the tour to the desired waypoint. Naviki then automatically adjusts the route to the new specification. The rules activated by the defaults continue to apply. You can find a list of all the waypoints inserted in this way in the history via the buttons. The waypoints can be moved or removed later using the

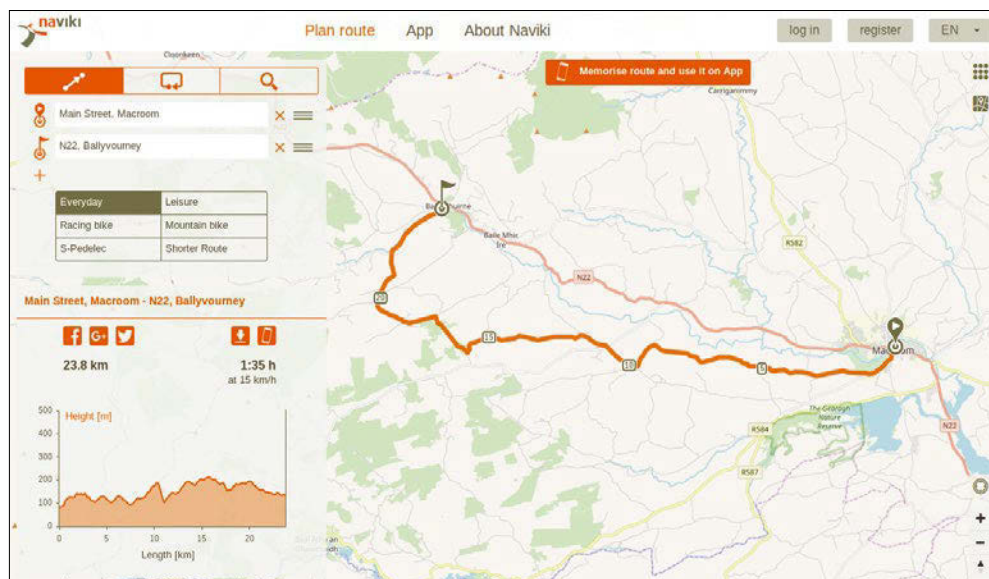


Figure 5: Naviki lets you quickly create routes for different bike types with a mouse click.

S-Pedelec

As one of only a few bicycle route planners, Naviki offers an option to create routes especially for S-Pedelecs. An S-Pedelec is an e-bike that can travel at speeds of up to 45kph. Traffic laws differ depending on the jurisdiction, but from a legal standpoint, an S-Pedelec is considered more like a moped than a bicycle.

route length, this option even integrates major roads into the route.

The other variants: *Racing bike*, *Mountain bike*, *Leisure*, or *S-Pedelec* (see “S-Pedelec” box) have appropriate effects. Each of these options implements a specific set of routing rules, so that the resulting routes can differ significantly in

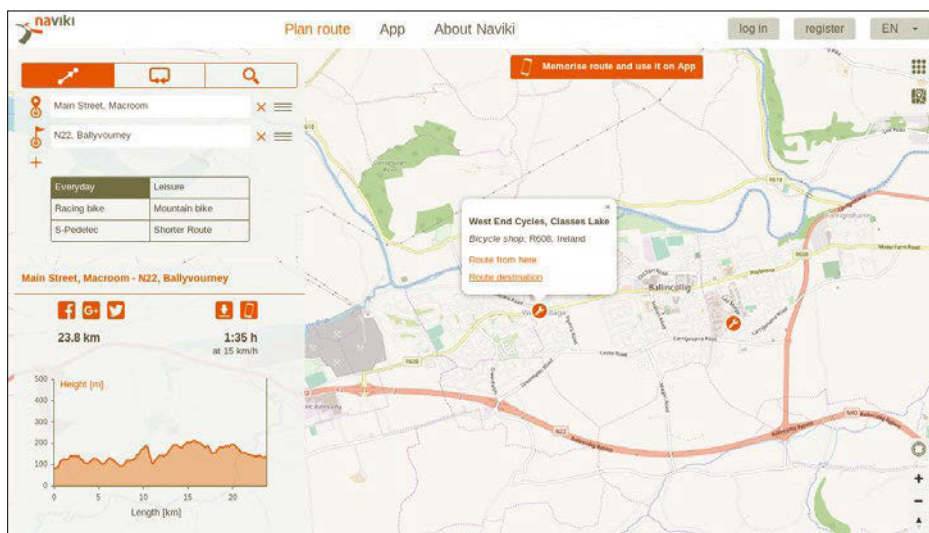


Figure 6: If required, Naviki displays useful points of interest, such as bike shops or inner-tube vending machines.

small x button to the right of each entry in the list.

Below the buttons for the route type is a fair amount of additional information: In particular, you will find the current route length and a height profile. If necessary, points of interest such as bike shops, inner-tube vending machines, or drinking water sources can be activated using the buttons in the top right corner of the map view (Figure 6). Use the button below to switch between the standard OSM, the OpenCycleMap, and a satellite view.

Once you are satisfied with the route, export the route. Click on the download symbol above the altitude profile. This symbol will open a dialog where you can choose the different formats. Naviki supports the most important data formats for navigation devices, including KML, GPX, OVL, and TCX.

OpenRouteService

OpenRouteService (ORS) [8] is similar to Naviki, but a little different and with different results. The procedure for planning routes is similar: First select the start and end points and the desired vehicle type from the context menu. You can add additional points to the route with a mouse click (Figure 7).

If necessary, the automatically selected route can be further adjusted

via *Options*. For instance, you can elect to avoid certain types of routes, ferry trips, or hills, or you can choose to limit the maximum gradient. You can also optimize the route selected by ORS by entering your own fitness level and top speed (Figure 8).

A real benefit of ORS is found below the route summary: An overview informs you about the road surface condition (Asphalt, Paved, Compacted Gravel, etc.), the way types (State road, Cycleway, etc.) or the steepness. If you mouse over one of the bars, ORS marks the corresponding sections on the map.

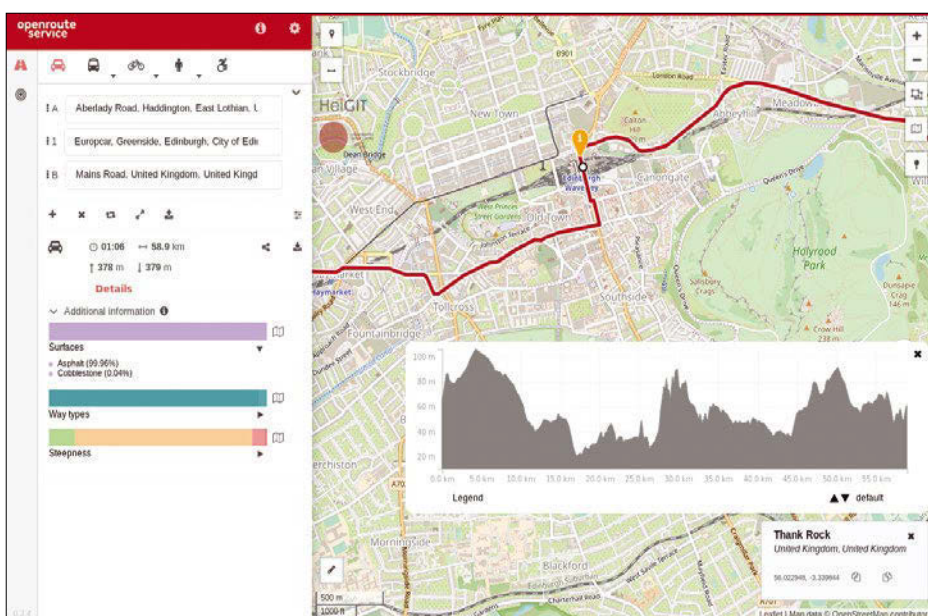


Figure 7: ORS creates routes for cars, bicycles, or pedestrians.

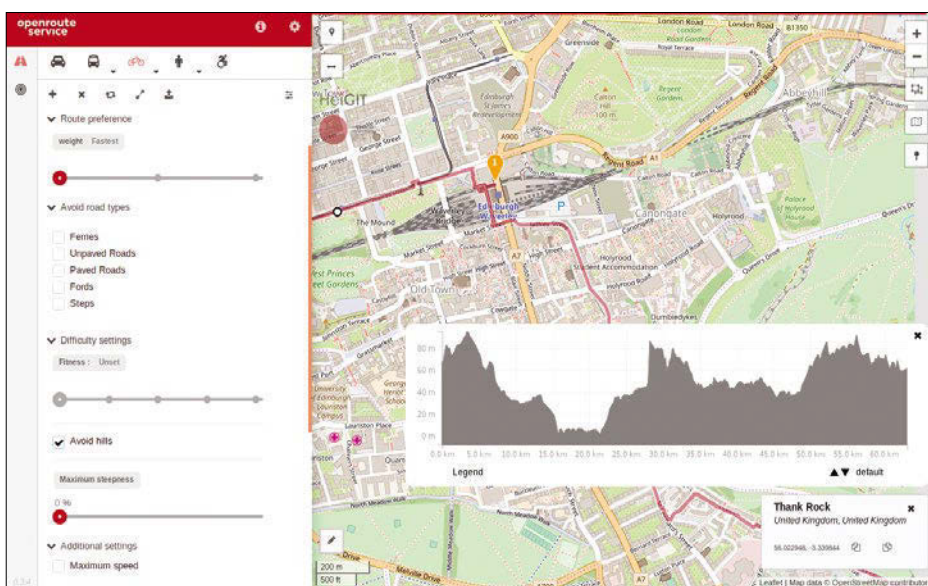


Figure 8: ORS lets you confine the route to a predefined maximum speed and gradient.



Listing 1: Overpass Query

```

01 nonumber
02 /*
03 This is an example Overpass query.
04 Try it out by pressing the Run button above!
05 You can find more examples with the Load tool.
06 */
07 node
08 [amenity=drinking_water]
09 ({{bbox}});
10 out;

```

This makes it easy to check whether the chosen route meets your own requirements – for example, riding a racing bike over rough gravel is not much fun.

You can save the results of your planning using the *Export Route* button, which can be found next to the route details such as travel time, distance, and altitude, in the form of a GPX track on your hard disk. Alternatively, the KML, GeoJSON, or JSON formats are also available.

Overpass Turbo

The OSM developers have continuously improved access to their database. One of the last and very successful enhancements to OSM is Overpass Turbo [9]. OSM calls Overpass

Turbo a data collection tool, but it is also a powerful tool for finding data in the database.

One special feature of Overpass Turbo is its impressive speed, which is the reason why they call it “Turbo.” You can create a query that you then send to the OSM database (Listing 1), and Overpass Turbo displays an interactive map with the hits (Figure 9).

The example given on the page searches the current map section for freely accessible drinking water points. You don’t necessarily have to familiarize yourself with programming the tool; a wizard helps you create the queries.

The Overpass Turbo query language contains some special elements: `{{bbox}}` stands for the currently displayed map area, and `{{date}}` allows

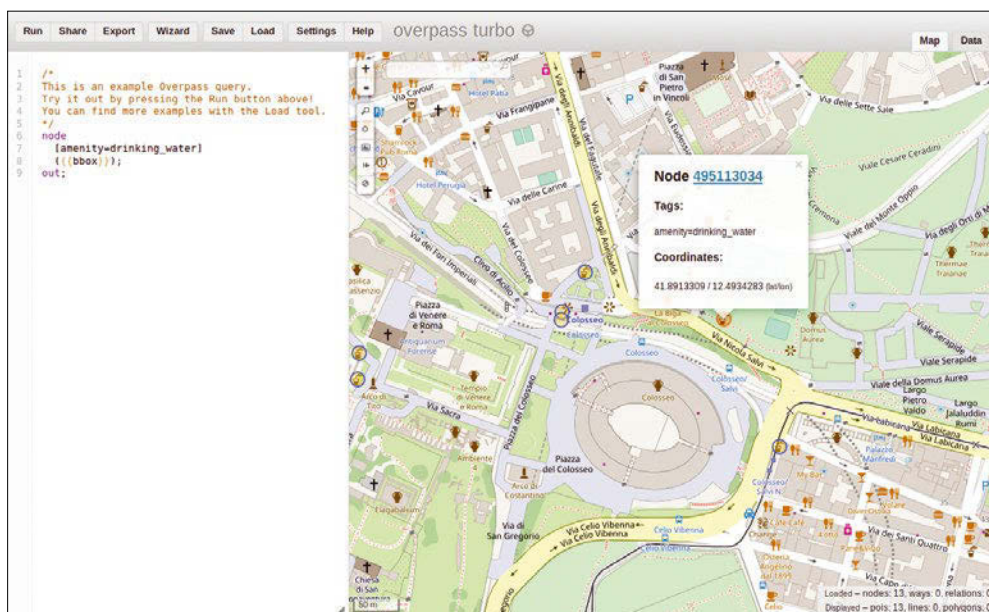


Figure 9: Overpass Turbo lets you search for a service or amenity and marks the matches with circles.

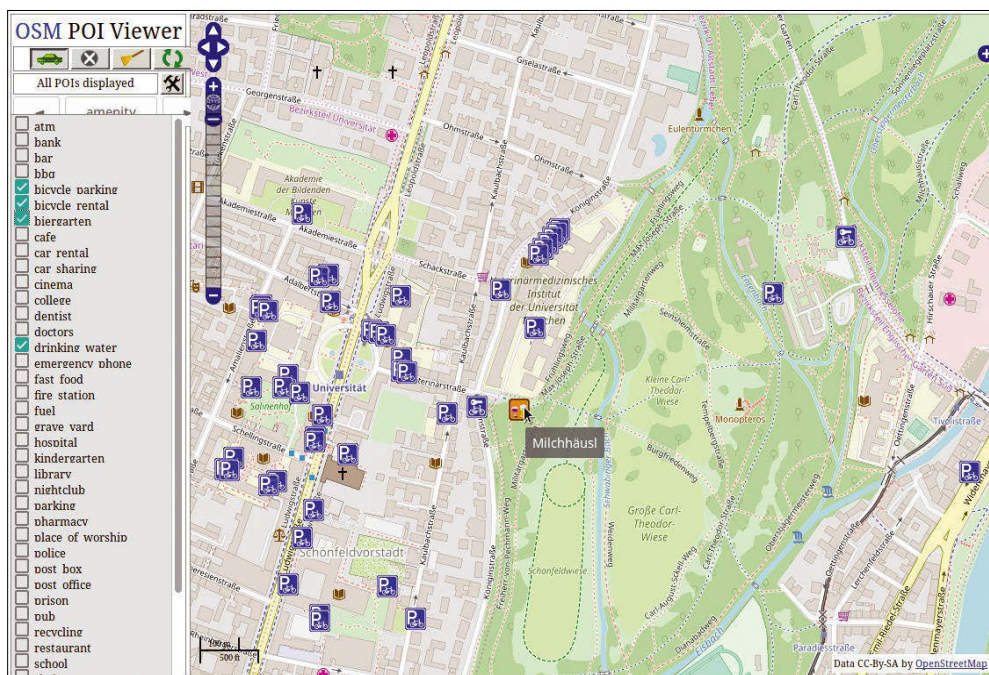


Figure 10: OSM POI Viewer helps you quickly find interesting points on the map, such as cycle parking lots and beer gardens.

OSM POI Viewer

The OSM POI Viewer [11] (Figure 10) offers a far simpler alternative to Overpass Turbo. The idea behind the POI Viewer is quite simple: On the left, first select the desired POI category from the sidebar, and then the page automatically starts the database query and creates a layer overlaid with the POIs. Depending on the size of the area displayed and the number of POIs selected, this can happen very quickly, or it can take a few seconds. The big advantage of this site is the very easy handling, but the rigid rules applied by the page limit your options.

MOBILE USERS

search for us today at your digital newsstand!



Only a swipe away!

Download our convenient digital editions for your iPad, iPhone, or Android device.

GOOGLE PLAY MAGAZINES

(UK AND AUSTRALIA)

ADMIN Magazine

Linux Magazine



APPLE NEWSSTAND

ADMIN Magazine

Linux Pro Magazine



Visit our apps page for more information: shop.linuxnewmedia.com/us/apps

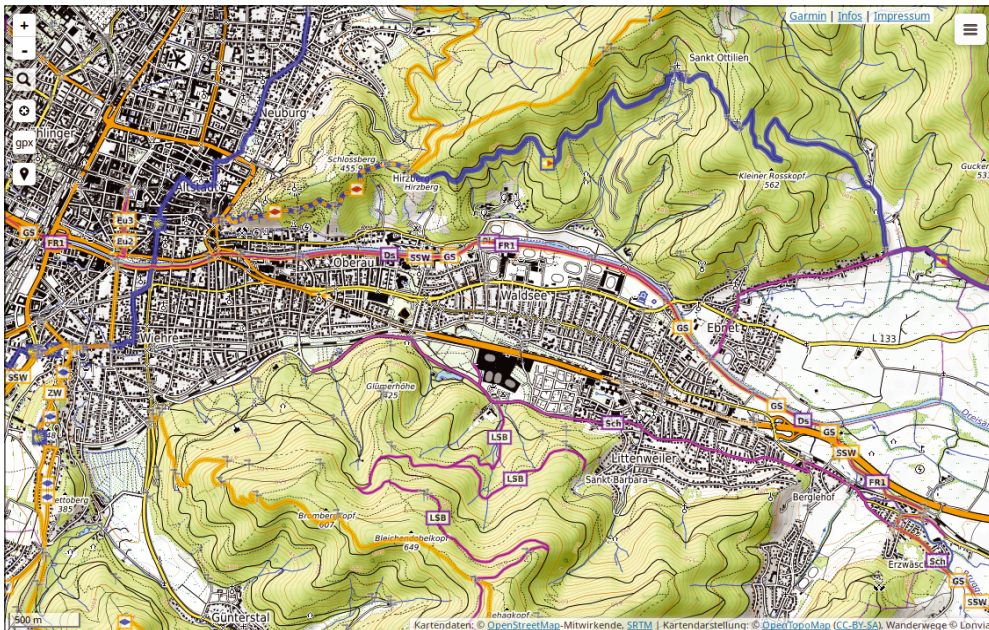


Figure 11: OpenTopoMap displays detailed contour lines.

time restrictions. There are a number of other ways to restrict the output. An article in the OSM Wiki describes the syntax of the queries in detail [10].

The page can also be accessed via a number of key combinations. Ctrl + Enter starts the current query, Ctrl + I opens the wizard, Ctrl + S saves a query, and Ctrl + O lets you reload stored queries. Ctrl + Z and Ctrl + Y (undo and redo) are available, as well as Ctrl + X, Ctrl + C, and Ctrl + V (cut, copy, and paste).

After you enter a query, Overpass Turbo displays a page with markers. This result can be saved in different ways using the *Export* button. In addition to a static output as a PNG image, Overpass Turbo also provides the formats XML, GPX, JSON, and GeoJSON. If the details of Overpass Turbo queries are too much trouble, you might want to consider OSM's POI Viewer (see the box entitled "OSM POI Viewer").

Topographic Maps

OSM includes current paths and roads, as well as outlines of cities, municipalities, or forests and rivers, but it lacks elevation data. The map service itself does not support contour lines or a topographical view; only summits and heights are recorded. Therefore, maps with terrain surfaces cannot easily be derived from OSM; you need to add data from other sources.

OpenTopoMap [12] provides a view of terrain surfaces (Figure 11). Other tools, such as HikeBikeMap [13]

and the Riding and Hiking Map [14], also provide topographical services, but they are primarily focused on Europe.

Comparing Maps

Geofabrik is one of the first commercial companies to work with OSM data and offer services such as software development, training, or consulting. In return, the company regularly provides copies of the OSM database for downloading [15]. In addition, Geofabrik also offers various tools, such as Map Compare [16], a tool that allows the direct comparison of up to four maps. If you move the view in one map, Map Compare

automatically loads the corresponding area in all the other maps (Figure 12). In addition to various modifications of OSM, you can also choose between Google Maps or Here Maps.

Conclusions

The excellent data material of OSM makes it attractive and relatively easy to set up services that extend OSM capabilities. Many of these services are freely available and compare well with commercial alternatives.

Sometimes it is not necessary or even useful to use preconfigured map services. In many cases, self-made maps provide better results. Open source tools available for Linux like Maperative [17] and QMapShack [18] help you build your own maps. ■■■

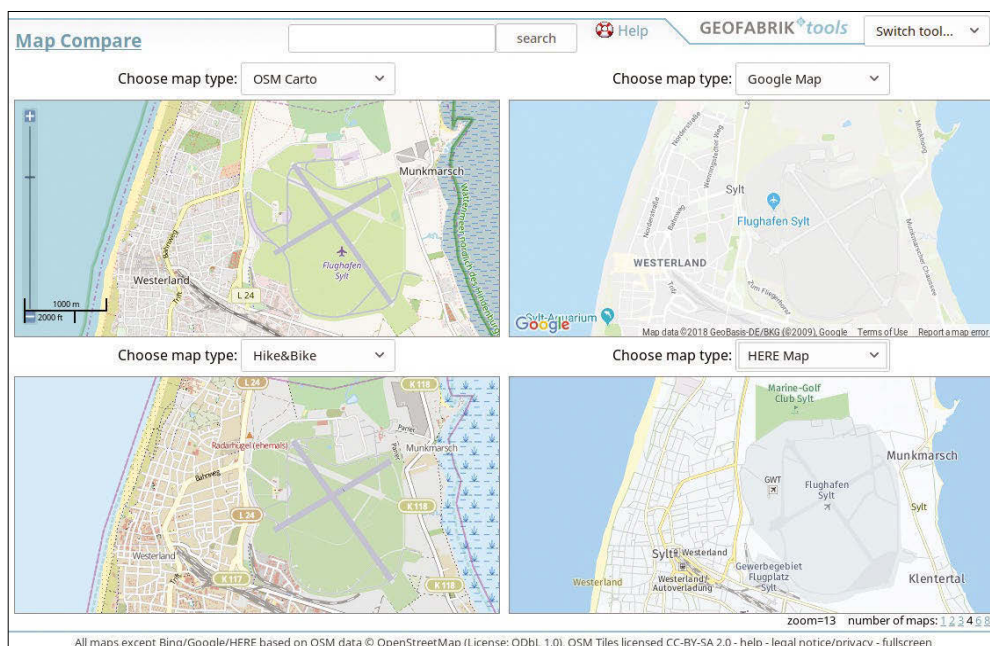
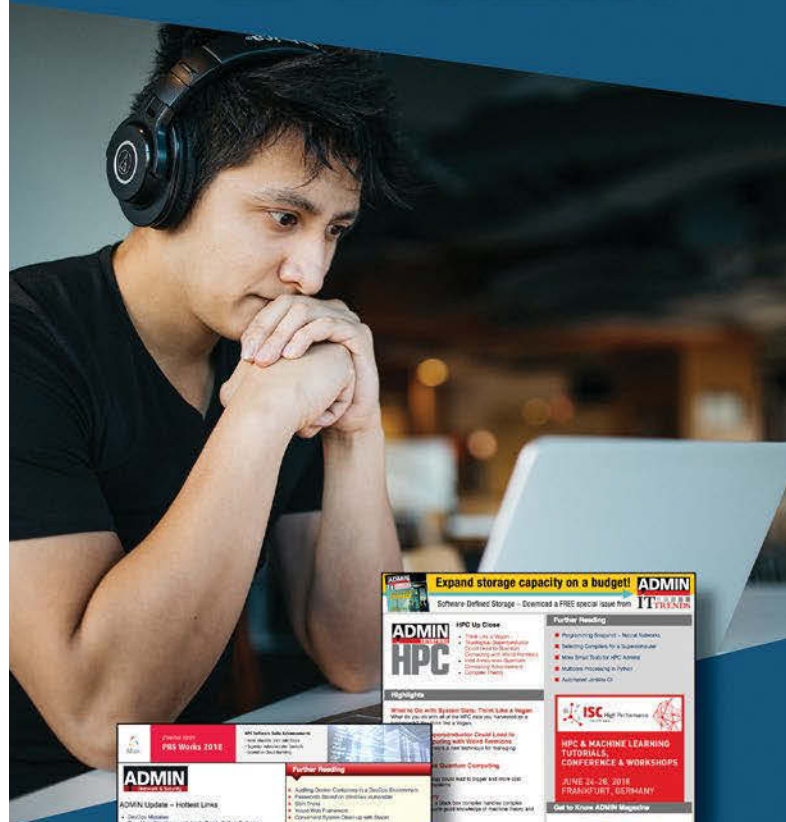


Figure 12: Geofabrik's Map Compare lets you compare up to four maps.

IT Highlights at a Glance

Info

- [1] OSM standard map: <https://openstreetmap.org>
- [2] Copyright and license: <https://www.openstreetmap.org/copyright>
- [3] OpenCycleMap: <https://opencyclemap.org>
- [4] North Sea Cycle Route 12: http://www.eurovelo.com/en/eurovelos/eurovelo-12?set_language=en
- [5] Waymarked Trails: <https://waymarkedtrails.org>
- [6] Routing services: https://wiki.openstreetmap.org/wiki/DE:List_of_OSM-based_services#Routing
- [7] Naviki: <https://www.naviki.org>
- [8] OpenRouteService: <https://openrouteservice.org>
- [9] Overpass Turbo: <http://overpass-turbo.eu>
- [10] Overpass Turbo syntax: https://wiki.openstreetmap.org/wiki/Overpass_turbo/Extended_Overpass_Turbo_Queries
- [11] OSM POI Viewer: <http://www.lenz-online.de/cgi-bin/osmpoi/osmpoi.pl>
- [12] OpenTopoMap: <https://opentopomap.org>
- [13] HikeBikeMap: <http://hikebikemap.org/>
- [14] Riding and Hiking Map: <https://www.wanderreitkarte.de/index.php?lang=en>
- [15] OSM data download: <http://download.geofabrik.de>
- [16] Map Compare: http://tools.geofabrik.de/mc/#13/52.6334/9.2331&num=4&mt0=mapnik&mt1=google-map&mt2=hike_bike&mt3=mapnik-german
- [17] Maperitive: <http://maperitive.net>
- [18] QMapShack: <https://bitbucket.org/maproom/qmapshack/wiki/Home>



GOT CLUSTER?

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

ADMIN HPC

HPC Up Close

- Let the Editor Wars Begin!
- Use Alliance Double-Checks
- Wing: Saves \$2 Million
- XSEDE Starts a New Era
- TKpart

Highlights

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

XSEDE Starts a New Era
NSF announces details for new XSEDE 3.0 phase

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

TKpart
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
Quickest 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 HPC Options for Tuning and Management
- New Service Will Adapt 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 PSE 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 next 45 days.

admin-magazine.com/hpc

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.

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

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

Linux Update:
www.linux-magazine.com/newsletter

ADMIN

Network & Security

COMPLETE ARCHIVE DVD
ISSUES 00-39

ISSUE 40/2017

While this ADMIN magazine disc has been tested and is to the best of our knowledge free of malicious software and defects, Admin Magazine cannot be held responsible and is not liable for any disruption, loss, or damage to data and computer systems related to the use of this disc.



OVER
3,000
PAGES OF UP-CLOSE
SYSTEM
ADMINISTRATION

Smart and Practical Information
for Network Professionals



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

A distro for moving from Windows/Mac OS to Linux

Migration

If you are looking to migrate from Windows and Mac OS to Linux, but still need Windows compatibility and applications, Zorin OS might be the solution. *By Swapnil Bhartiya*

With its release of Zorin OS 12.4 in August, Zorin OS [1] positions itself as an alternative to Windows and Mac OS, “designed to make your computer faster, more powerful, and secure.”

When a distribution makes such a claim, it could mean several things. Most distributions stop at skinning the theme to resemble Windows XP, without actually bringing in support for the Windows ecosystem.

Users are attracted to such distributions, but it leads to buyer’s remorse when they cannot bring their existing workload to Linux.

I have been migrating users from Windows to Linux for the last 13 years. I moved my wife to Ubuntu and eventually Chromebook. In my experience, what really helps people migrate from one platform to another goes beyond look and feel. It’s more about ease of use and the ability to bring your existing workload to the new platform.

I looked at Zorin OS from three perspectives: support for Windows applications, compatibility with Windows, and ease of use.

A Doorway into the Windows World

Since Zorin specifically courts Windows and Mac OS users, it has created tools

that can mimic Windows 10 or Mac OS layouts with one button. The idea is not to copy the layout but to offer a familiar interface that makes users comfortable.

The paid version, Zorin OS Ultimate (more discussion about this version later), comes with the Zorin Appearance tool, which includes six layout options: Windows (two layout options), Gnome 2, Ubuntu Unity, Gnome 3, and Mac OS (Figure 1).

“If you’re familiar with Windows, everything will be right where you remember it – from the start menu to the taskbar – so you don’t need to learn anything new,” explained Artyom Zorin, co-founder and CEO of Zorin Group. “If you are more familiar with Mac OS, you can open the Zorin Appearance app to change the entire desktop layout to resemble Mac OS in a click.”

Unfortunately, Zorin OS doesn’t include icon themes that resemble those themes found in Windows 10 or

Mac OS, apparently due to copyright and legal issues. However, users can safely download and install the desired themes from Gnome-Look.org. Just extract the downloaded .zip file and copy the extracted folder to /usr/share/icon and /usr/share/themes:

```
# rsync -avzP /home/swapnil/Download/theme.zip /usr/share/themes/
# rsync -avzP /home/swapnil/Download/theme.zip /usr/share/icons/
```

Then, change the theme with the Zorin Appearance tool or Gnome Tweak Tool. It’s that simple.

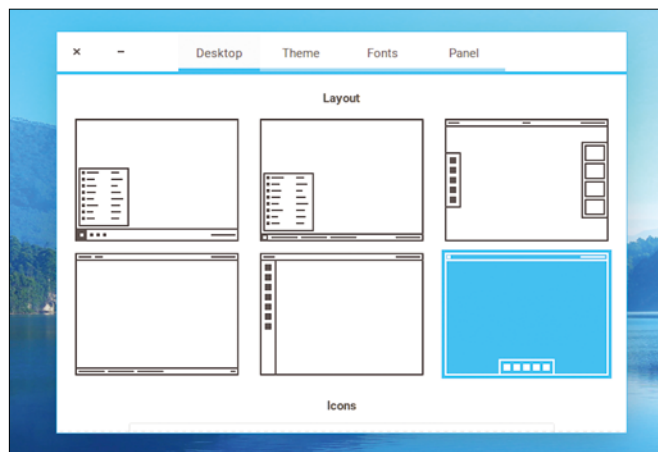


Figure 1: Zorin OS Ultimate edition offers customized layouts to mimic the look and feel of Mac OS and Windows.

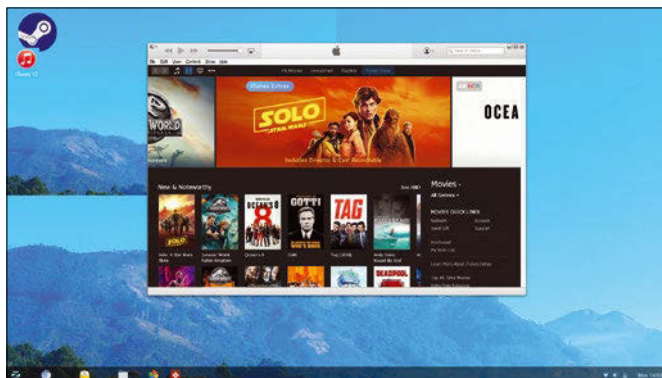


Figure 2: iTunes running in Zorin OS using PlayOnLinux and Wine.

Windows Apps on Zorin

A Windows or Mac OS user won't be content with just the familiar look and feel. One of the reasons people choose a platform is the availability of the apps they use. It could be just one app that they use once in a while; it could be Microsoft Office or some tax app.

Zorin OS comes with PlayOnLinux, a third-party front end for Wine, that makes it easy to install and manage supported Windows applications in Zorin. Wine has a very comprehensive catalog of supported Windows apps.

Installing any Windows 10 app on PlayOnLinux is very easy. Open PlayOnLinux and search for the desired application, such as iTunes 12. Click on the *Install* button and PlayOnLinux will download all required packages and create a virtual drive for iTunes in Wine. PlayOnLinux doesn't offer the actual app, as this could lead to legal issues. You need to provide the official app's executable binary, which you can easily download from the official source. (A caveat: Never download apps from random websites.)

I don't rely on Windows 10 apps, but for the sake of testing, I tried iTunes. iTunes 12.13 is the supported version, which you can download from Apple [2].

When prompted by PlayOnLinux, just point to this .exe file. It will open Wine, where you select Windows 10 as the target platform. Click *Next* on PlayOnLinux, and you will have iTunes running on your machine. Take a moment and think about it: You are running an app created by Apple for Windows 10 in Zorin OS (Figure 2).

PlayOnLinux/Wine, in my experience so far, is a good solution for running certain legacy applications that people still

need. It's certainly not capable of installing modern applications like Adobe Creative Cloud suite, but Wine is getting better with each release.

Valve Software, the company behind PC game distributor Steam, has put its weight behind Wine and created a modified version of Wine called Proton [3] to bring popular Windows games to Linux. Distributions like Zorin OS benefit from this work as it means support for more Windows apps. It also means Windows users can now run some of their favorite games in Zorin OS!

"The Wine team has put in a lot of effort into supporting apps like Microsoft Office 2013 in the latest versions, and Office 365 looks like it'll have great support in Wine 4.0. This release cycle is getting amazing new support for Windows games since it was recently revealed that Valve Software has been funding the Wine project this year," said Zorin.

The Cloud's Silver Lining

The good (or bad depending on how you look at it) news is that most applications are moving to the cloud, and the cloud is OS agnostic. Linux users can enjoy the same cloud experience they get on Windows 10 and Mac OS.

All cloud-based applications, including Office 365, Adobe Lightroom, Apple's iCloud, and Apple Pages, are accessible in Zorin OS, without a workaround (Figure 3).

The Linux Experience

You've made the switch from Windows or Mac OS to Zorin OS, but how good is the Zorin experience, beyond appearance and support for Windows apps?

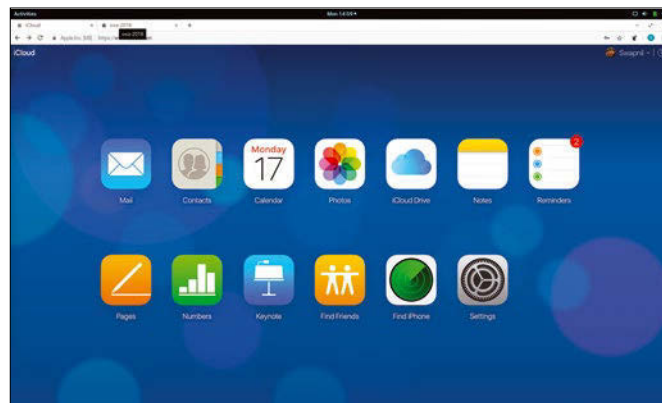


Figure 3: Cloud-based apps, like Apple's iCloud, run on Zorin OS without the need for a workaround.

Zorin OS uses Ubuntu LTS as the base, which means one of the most user-friendly distributions is at the bedrock of Zorin.

The first step into the Linux world is the installer. No matter how good an OS is, if you can't install it, nothing else matters.

Zorin OS offers one of the easiest installers, Ubiquity. It's simply a matter of clicking *Next*. You won't have issues as long as you are installing a single boot machine. However, if you want to dual boot with Windows 10 or another Linux distribution, Ubiquity provides an option to install Zorin OS alongside the existing OS or wipe the entire hard drive and install it as the solo OS. You can also enable options to install third-party software or download updates during installation. Zorin OS will install media codecs and additional drivers depending on your hardware (Figure 4).

Ubiquity does a lot of spoon feeding. However, if you are well-versed with Linux, then you can choose the *Something else* option and fine-tune your installation media as you please. It's designed for consumers without dumbing it down for advanced users.

Once the distro is installed, you boot into a very polished distribution (Figure 5).

Zorin OS targets [5] Windows and Mac OS users, but it's a very powerful Linux distro in its own right. In a nutshell, Zorin OS is a full-fledged Ubuntu-based distribution that adds another layer of ease of use and polish to Ubuntu.

"While Windows users are our main audience, we work to make Zorin OS an en-

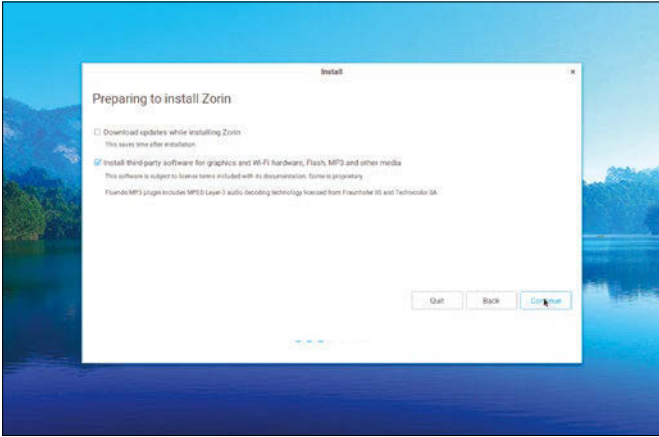


Figure 4: The easy-to-use Zorin OS installer.

joyable and accessible desktop for many other types of users as well,” said Zorin. “When we go about building a new release of Zorin OS, we start by defining the kind of user experience we want to deliver, and then work our way backwards to the features and technologies, rather than the other way around. This process encourages us to think of how we can get the technology out of the way, so you have a pure and frictionless computing experience that works the way you do.”

As a distro based on Ubuntu LTS, the strength of Zorin OS is familiarity, stability, and ease of use. Zorin OS’s user interface (UI) offers the familiar start menu UI where you click on the icon at the bottom left to open the application launcher (Figure 6).

The launcher’s left panel shows a selection of tools and applications; the launcher’s right panel shows the username and file directory, as well as quick access to *Software* (the application store), *Settings*, and *Activities Overview* (aka the dashboard).

Instead of using vague words and jargon, Zorin OS uses simple terms. A user

can easily guess that *Graphics* is for image editing software and *Office* is for productivity suites.

The app launcher also has a search bar, which makes searching a pleasure. A new user can simply type “editor” or “writer” to find the gedit text editor or LibreOffice Writer. A search for an image editor would offer Gimp.

Zorin OS comes with a decent set of preinstalled applications, including Firefox, Chromium web browser, LibreOffice, Gimp, Videos, Rhythmbox, and so on. If you need applications like Google Chrome or VLC, you can easily install them. There are many options for doing this:

- With Gnome Software tool, which comes with Zorin OS, you can easily search for the desired application and install it.
- If you are comfortable with the command line interface (CLI), you can easily install an application by running the `apt-get install` command.
- If an application is not available in *Software*, you can simply download the `.deb` file and install it with a double-click.
- You can add a package’s repository and install it from *Software* or the CLI. Using a re-

pository, if available, is recommended, as it will also install any future updates when you run the system update command.

- You can also install snap packages on Zorin OS.

Manual Package Installation

If you want to manually install a package in Zorin, it’s very easy. Google Chrome, for example, was not available in Zorin’s repository, so I headed to Google [4] and downloaded the application’s 64 bit `.deb` version. The installer was downloaded to the *Download* folder. Just double click on the file and install it as you would on Windows and Mac OS – easy peasy.

If you are comfortable with the command line, you can add Google Repository [6] to Zorin OS and install other Chrome and Google packages as well (Figure 7):

```
# sudo wget -q -O - https://dl.google.com/linux/linux_signing_key.pub | sudo apt-key add -
```

To refresh repositories, run this command:

```
# sudo apt-get update
```

Now you can install all Google packages with:

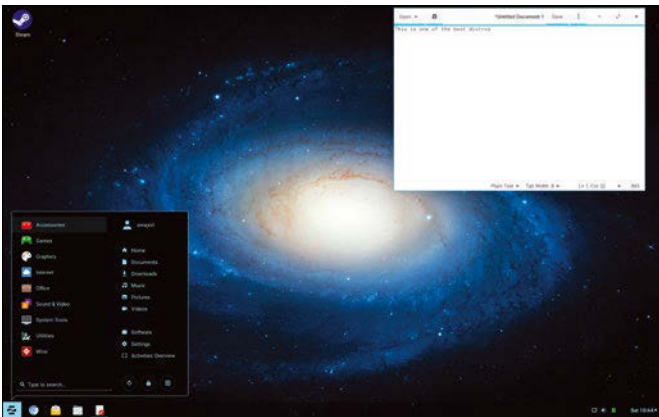


Figure 5: The default look-and-feel of Zorin OS.

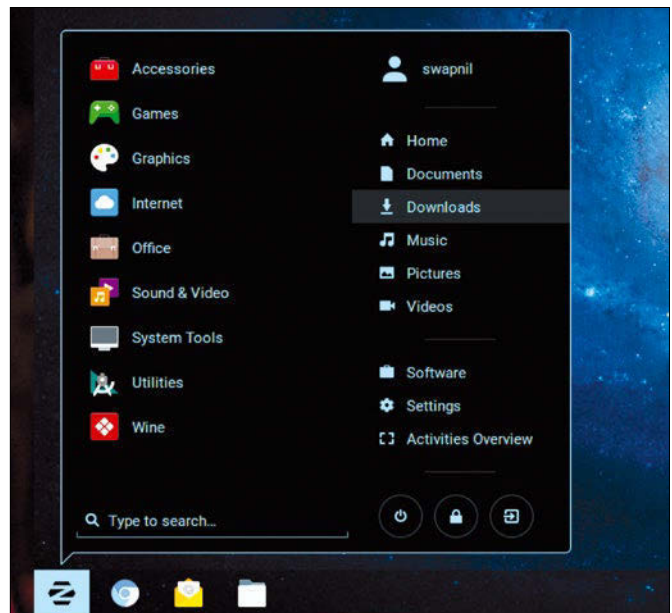


Figure 6: Zorin OS’s application launcher offers a familiar interface for most PC users.

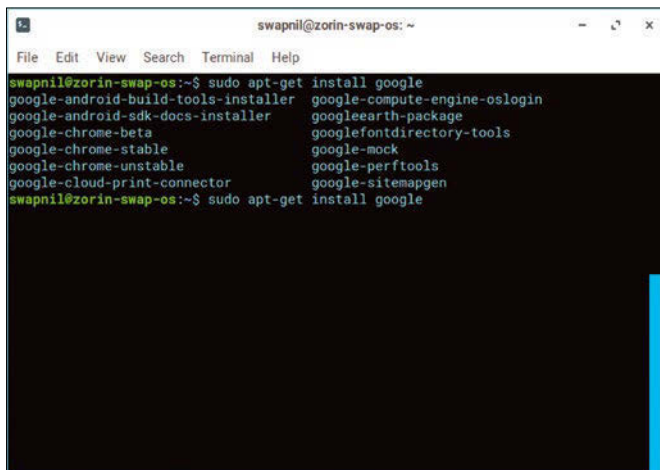



Figure 7: Once you add Google Repository, you can install all Google packages available for Zorin OS.

```
# sudo apt-get install 
google-chrome-stable
```

Updating and managing the system is also very easy. The Software tool is capable of not only updating the applications, but it can also install OS updates (Figure 8). Since it's Ubuntu, you can always run the following command to keep the system updated manually:

```
# sudo apt-get update && dist-upgrade
```

Zorin vs. Zorin

Despite being a community-driven distro, Zorin OS also offers a business model. There are three versions of Zorin OS: Ultimate, Core, and Lite. Lite is free of cost and intended for low-powered and older systems.

Zorin OS Core is intended for modern systems. It's available for free, with an optional donation to support the project. Zorin OS Ultimate, the paid version,

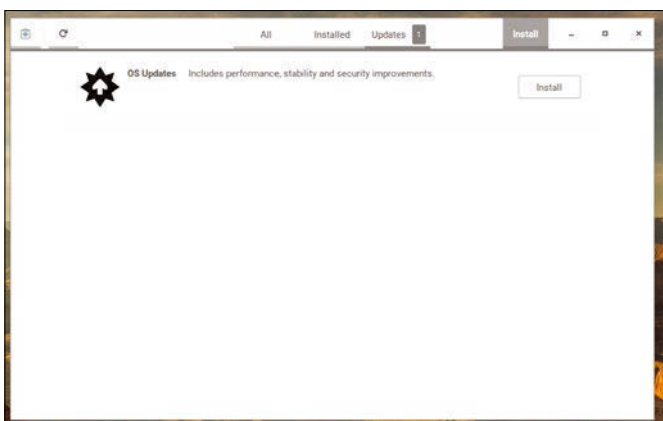


Figure 8: The Software app offers system-wide updates.

costs EUR19 and comes with additional features, official technical support, and more UI tweaks.

While the base of all three versions is the same, Zorin OS Ultimate includes a large selection of extra software including games, multimedia apps, and business software. It comes pre-

loaded with music production workstations, 3D modeling apps for creating professional movies and games, CAD software, accounting and finance tools, media center software, and a selection of native Linux games.

Zorin OS Ultimate also comes with Zorin premium software such as Zorin Background Plus and the extended Zorin Appearance (which I mentioned earlier). Zorin Appearance in Zorin OS 12 Ultimate includes six layouts as opposed to the three available in the Core edition.

The good news is that Zorin doesn't limit or restrict functionality in the Core edition. You can use Zorin OS like any other Linux distribution without paying a dime. You can always do the extra work of fine-tuning by googling and installing apps. However, if you plan to deploy Zorin OS in your organization, paying EUR19 might make more business sense.

Zorin also has customers in the enterprise and public sector. To help these customers, Zorin is now consid-

ering the entire life cycle of deploying, managing, and maintaining fleets of Zorin OS-based computers.

"We're in the process of building a remote management suite with the goal of managing tens, hundreds, or even thousands of workstations.

We're planning to release this product before the end of this year," said Zorin.

Conclusion

In a nutshell, Zorin has something for everyone – almost everyone. Whether you are a casual Linux gamer, a casual PC user, an enterprise customer, a Windows user on the fence, or an enthusiast like me, Zorin is worth trying out.

"Our developers and engineers are the same people who read and reply to customer support messages, which gives the whole team a deep understanding of what our users want out of Zorin OS and what parts of the operating system they might be having issues with. This knowledge goes straight back into the development process and helps us prioritize which features would help the user most and provide the best user experience possible," said Zorin.

As a result, Zorin OS is designed based on what users need and not necessarily what developers want.

I must admit that I was not previously a huge fan of Zorin. I always disliked distros that tried to mimic the appearance of Windows to appeal to those users. The distros didn't go beyond that, and the access to Wine and PlayOnLinux for running Windows applications never seemed as smooth as advertised.

However, ever since I actually used Zorin, I have become a huge fan of this distro. It does extra work to offer support for Windows applications. Even if you don't use Windows applications, Zorin OS is a great distribution for a Linux puritan. It has now become one of my most recommended distributions. Try it; you won't be disappointed. ■■■

Info

- [1] Zorin OS: <https://zorinos.com>
- [2] iTunes 12.13 download: https://support.apple.com/kb/dl1614?locale=en_US
- [3] Valve Software's Proton: <https://www.winehq.org/news/2018082301>
- [4] Google Chrome download: <https://www.google.com/chrome/>
- [5] Zorin OS targets: <https://www.linux.com/news/zorin-os-12-passes-one-million-downloads-mark-60-are-windows-and-mac-users-0>
- [6] Google Repository: <https://www.google.com/linuxrepositories/>



Deleting the old kernels lost on your hard drive

Spring Cleaning

When you update the kernel, the old version remains on the disk. If you clean up, the reward is several hundred megabytes of free disk space. *By Roman Jordan and, Christoph Langner*

All is flux; nothing stays still. This adage also applies to long-term Linux versions such as Ubuntu LTS and CentOS. The most important reasons for regular updates are security and troubleshooting. And the kernel develops, too. Recent patches for Spectre and Meltdown are good examples of important kernel changes that had to arrive through system updates.

But what happens to the old kernel after an update? First of all, the kernel is a normal software package. It is thus

Turning Back the Clock

The Linux kernel is responsible for many tasks, including managing hardware resources, enabling access to individual system components, and providing interfaces for system communication. Various hardware embodiments can sometimes impair this harmonious relationship. In the worst case, the system will not boot after a kernel update. If there are problems after the update, the old kernel might help, at least until the next kernel update. For this reason, most distributions keep one, but more likely several, of the previous kernel packages on disk.

managed through the package management system for your Linux variant. In the case of a Debian or Ubuntu system, this means Apt; in the case of Red Hat-based systems, RPM and its friends Yum and DNF.

If you take a closer look, you'll see that there is no update for the kernel package: The package manager simply adds the new version to the system without deleting the previous version. Often several old versions will remain on the disk (see the box "Turning Back the Clock"), which requires a large amount of disk space.

This article shows how to remove old kernel images from your system. The reference systems are Debian 9, Fedora 28, Ubuntu LTS 16.04, and CentOS 7.5. By default, these distributions limit the number of kernel versions installed on the system – at least if they have been installed via a package update. For manually installed versions, or for additional kernel-related packages, the techniques described in this article might not apply (see the box entitled "Kernel-Related Packages").

For the purposes of this article, the term "kernel package" refers to all packages belonging to the kernel, including

kernel-related packages. As Figure 1 shows, these packages occupy a significant amount of disk space.

Manual Removal

Handling the software installed on the system is the package manager's task. The basic procedure is similar for all distributions: First, determine the version of the running and working kernel. You

Kernel-Related Packages

Kernel-related packages are packages with files that are directly related to the version of the kernel. These files include, for example, the kernel modules, the RAM disks used at boot time, the kernel headers, and sometimes a few add-ons that depend directly on the kernel version. Which of these components are in which packages is determined by the distribution.

The software manager remembers which packages are installed. For example, if you have installed the kernel headers, the new package with kernel headers will be installed on your disk each time the kernel is updated. The system keeps the previous version along with the actual kernel.

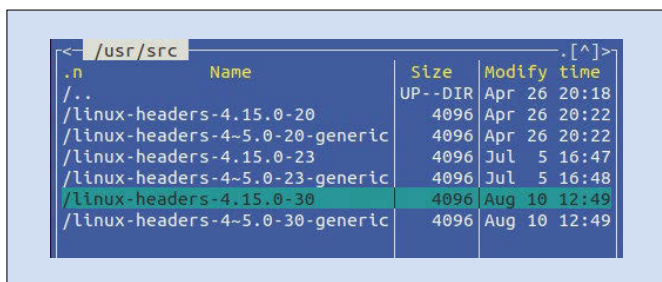


Figure 1: The kernel headers also need space (Debian).

need to be sure this current version stays on the system. Then display all kernel packages. The last step is to delete the packages that are no longer required. (Be sure you observe the instructions in the “Caution” box.)

Determining the Current Version

The GRUB bootloader usually displays a selection of installed kernel versions at boot time. For Red Hat-based versions, this output is direct; for Debian the list appears in the *Advanced Options* area (Figure 2); for Ubuntu, you need to press the Shift key to see the list.

The `uname -r` command, which is available with all distributions, displays the version of the running kernel. The output for a Debian system is as follows:

```
$ uname -r
4.9.0-6-rt-amd64
```

Package Service

Distributions differ in the name and contents of kernel packages. Table 1 gives

an overview of the commands you might need to determine the package names, organized by distribution. The commands listed in the table provide the complete names. The names of the packages alone indicate whether they are kernel packages or components.

Making Space

Once you have completed the preparations, you can start removing the packages you no longer need. Thanks to the autocomplete function with the

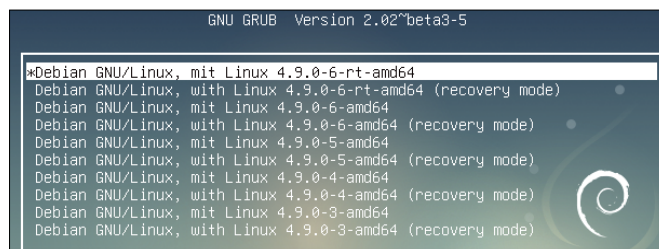


Figure 2: Debian provides a complete list of installed kernel versions in the Advanced Options area.

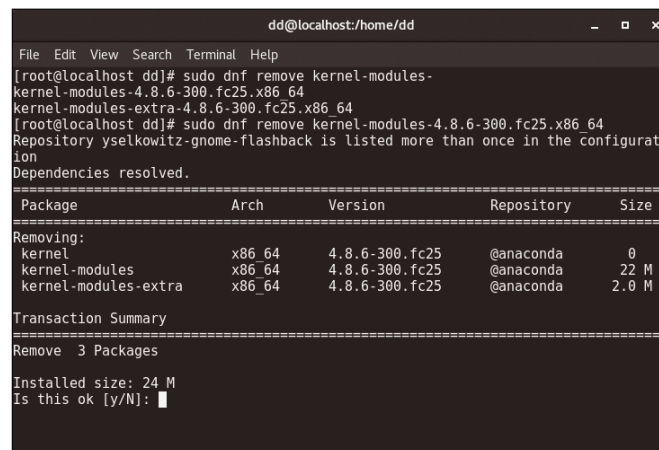


Figure 3: In Fedora, deleting the modules will delete more packages.

Caution

Problems and data loss can occur when using the package management system. Therefore, make a backup before deleting old kernel versions.

Some distributions even let you delete the current kernel version. Please pay attention to the screen output. Do not delete the last working kernel. Uninstall older kernel versions only if the system is working perfectly.

Use the abstracted commands of the package manager, for example the `yum` command instead of `rpm` for CentOS. The abstracted commands display an overview before the actions are executed and let you cancel.

Table 1: Determining Package Names

| Distribution | Command | Examples of Package Names |
|--------------|---|--|
| CentOS | <code>rpm -qa "kernel*"</code> | <code>kernel-3.10.0-862.9.1</code> , <code>kernel-headers-3.10.0-862.9.1</code> , <code>kernel-devel-3.10.0-862.9.1</code> |
| Fedora | <code>rpm -qa "kernel*"</code> | <code>kernel-4.16.3-301</code> , <code>kernel-modules-4.16.3-301</code> , <code>kernel-devel-4.16.3-301</code> |
| Debian | <code>apt list --installed linux-*</code> | <code>linux-image-4.9.0-6-amd64</code> , <code>linux-headers-4.9.0-4-amd64</code> |
| Ubuntu LTS | <code>apt list --installed linux-*</code> | <code>linux-image-4.15.0-30-generic</code> , <code>linux-headers-4.15.0-30-generic</code> , <code>linux-image-extra-4.15.0-15-generic</code> |

Table 2: Removing Kernel Packages

| Distribution | Command | Sample Command | With Wildcard |
|--------------|--------------------------------------|---|---|
| CentOS | <code>yum remove package_name</code> | <code>yum remove kernel-3.10.0-693.2.2 kernel-devel-3.10.0-693.2.2</code> | <code>yum remove "kernel*3.10.0-693*"</code> |
| Fedora | <code>dnf remove package_name</code> | <code>dnf remove kernel-4.16.16-300 kernel-core-4.16.16-300</code> | <code>dnf remove "kernel*4.16.16-300*"</code> |
| Debian | <code>apt purge package_name</code> | <code>apt purge linux-image-4.9.0-3</code> | <code>apt purge linux-*4.9.0-3-rt*</code> |
| Ubuntu LTS | <code>apt purge package_name</code> | <code>apt purge linux-image-4.9.0-3</code> | <code>apt purge linux-*4.9.0-3-rt*</code> |

tab key, which also works when entering the package name, and the typical Linux copy-paste function with the middle mouse button, handling the long file names is no problem.

Table 2 shows some commands for deleting kernel packages. As with all active package management activities, you need root privileges on the system to delete kernel packages.

The fact that the kernel packages within a distribution typically start with a uniform name, and that the commands allow wildcards, means that you can delete all the kernel packages of a version with a shortcut. The last column of the

table shows some examples. If necessary, add more packages or wildcards to the call.

The commands shown in Table 2 provide an overview of the subsequent actions. Pay special attention to this output if you are using wildcards, and cancel the action if necessary. All commands let you specify multiple packages. Removing a kernel version also deletes the corresponding entry in the boot menu.

Some distributions draw “logical conclusions” and sometimes delete more packages than specified. For Fedora, deleting the kernel modules or kernel core packages also deletes the kernel (Figure 3).

Some of the distributions have their own tools for handling the kernel packages. The following sections describe a few of those tools.

Fedora and CentOS

By default, Fedora limits the number of installed kernel packages to three; CentOS sets the limit at five. The value of the variable `installonly_limit` determines the number. Fedora understands `installonly` packages to mean packages that can be updated via the package manager, but the old version is not deleted until this limit is reached.

Fedora sets this `installonly_limit` value in `/etc/dnf/dnf.conf`; CentOS sets the value in `/etc/yum.conf`. If you leave this value at 2 or higher, a fallback kernel always remains on the disk [1]. To edit these files, you need root privileges [2]. Be sure you make a backup copy before you make any changes.

Using the `Package-config` tool from the `yum-utils` package, CentOS gives you a simple approach to deleting previous kernel packages. The `yum-utils` package is installed by typing `sudo yum install yum-utils`. The `sudo package-cleanup --oldkernels --count=2` command deletes all kernel packages except the last two.

Debian and Friends

The Synaptic program is part of many Debian-based distributions. It is installed

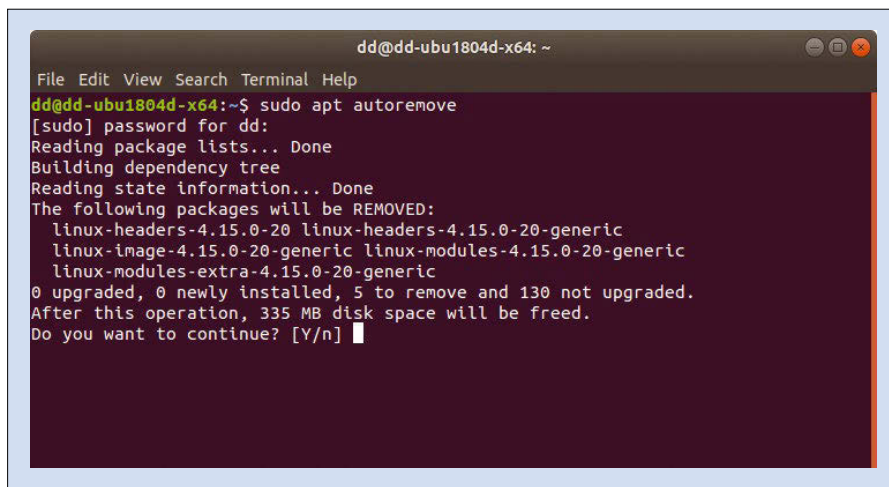


Figure 4: The `apt autoremove` command removes unneeded packages from the hard disk on Debian and its descendants.

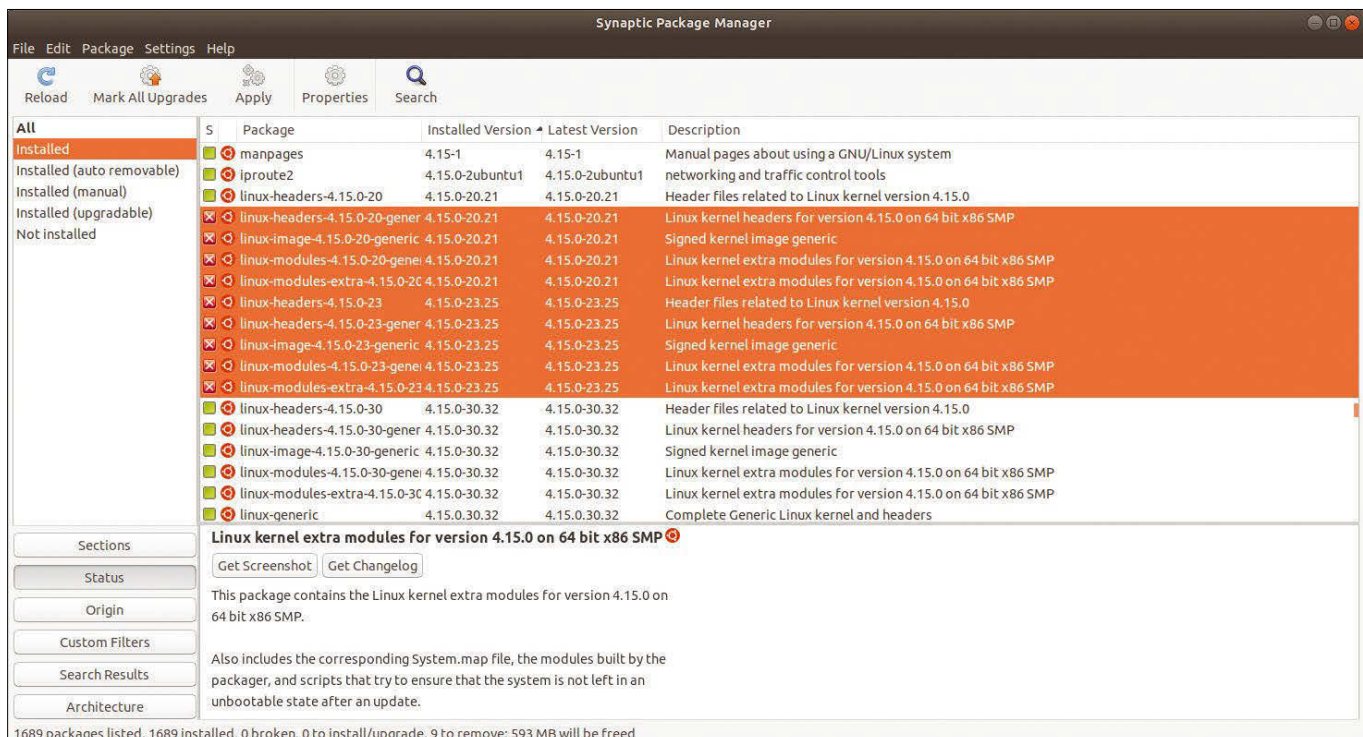


Figure 5: Graphical software management with Synaptic.

Debian Automatic

Where software requires further packages as dependencies (e.g., libraries for extended functions), the package manager automatically includes them when installing the application. If you remove the program from the system again, these additional “orphaned” packages often remain on the hard disk without serving any useful purpose.

The package managers on all Debian-based distributions (including Ubuntu or Linux Mint) have a routine with the `sudo apt autoremove --purge` that lets you remove this unnecessary ballast from the system. In addition to the orphaned packages, the command also cleans up the kernels (Figure 4). This automatic procedure keeps the current and the second-to-last kernel on the system.

If the routine ignores the obsolete kernels, it may be because they are tagged as “manually installed.” The `apt-mark showmanual | grep linux-image` command lists the packages (and filters out the kernel packages immediately). If you then change the status by typing `apt-mark auto package_name`, the caretaker should now clean up the kernel packages as well.

by typing `sudo apt install synaptic`. Depending on the distribution, the call is either `synaptic` or with `synaptic-pkexec`. Alternatively, you can launch Synaptic via the application menu.

Here too, it is a good idea to first determine the version of the currently running kernel by typing `uname -r`. To quickly access the kernel packages, launch Synaptic and set the *Status* to *Installed* in the left part of the window, click to activate the area where the packages are listed, and type *linux*.

Mark the packages (`*image*`, `*header*`, etc.) of the old versions and select the entry *Mark for complete removal* in the context menu (right mouse button, Figure 5). Clicking on *Apply* starts the de-installation. See the box entitled “Debian Automatic” for additional information.

Conclusions

Automatic management of the kernel packages is typically fine for normal use, but automatic systems sometimes show some weaknesses. Manually installed kernel packages are usually not

governed by the automatic mechanisms. If you manually install kernel packages on the disk, it makes sense to search the system for obsolete packages from time to time. If you need disk space in a hurry and can do without the previous kernel versions, removing large kernel packages will help in the short term. ■■■

Info

- [1] “Configuring DNF and DNF Repositories”: https://docs.fedoraproject.org/en-US/Fedora/24/html/System_Administrators_Guide/sec-Configuring_DNF_and_DNF_Repositories.html
- [2] “Configuring Yum and Yum Repositories”: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/deployment_guide/sec-configuring_yum_and_yum_repositories

Author

Roman Jordan has been working with Linux for over 20 years. His main focus is on the Linux kernel and on programming small embedded platforms.

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

GET IT NOW!

SAVE TIME ON DELIVERY WITH OUR ALTERNATIVE PDF EDITIONS




```
body {
  font: x-small;
  background: #;
  color: black;
  margin: 0;
  padding: 0;
}
```

Making the most out of Cascading Style Sheets

CSS Basics

Cascading Style Sheets can provide a sophisticated layout for articles and books. Here are a few tips to get you started. *By Bruce Byfield*

If you have created HTML or JavaScript in the last two decades, you probably have a nodding familiarity with Cascading Style Sheets (CSS) [1]. Not only is the CSS stylesheet language commonplace on the Internet, but popular ebook formats like EPUB and MOBI use it as well. Moreover, because CSS is human-readable, if you know English, you can understand much of what is being done without learning the details. However, a formal look at CSS is likely to reveal missed details, not least of which is the number of options it provides.

CSS keeps format and content separate. Initially, markup languages like HTML were designed to do the same, but the distinction became blurred (if you are old enough, you may remember the infamous HTML blink tag). CSS was intended to restore the distinction by being a stylesheet

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>

that gave designers control over how pages are viewed. CSS has been only partly successful, because different web browsers continue to display some elements differently and to allow readers to override CSS with their own preferences. All the same, CSS remains popular, if only because most designers prefer to have some formatting control.

At any rate, CSS continues to be a standard feature of modern computing. Its specifications are developed and maintained by the World Wide Web Consortium (W3C), which also provides a CSS validation service [2]. However, be aware that different CSS uses, such as ebook publication on Amazon or Lulu, often have their own, sometimes arbitrary, standards for CSS stylesheets. A given stylesheet can be valid by W3C standards yet still fail to be accepted by a given publisher. It's time, though, for a closer look at CSS constructions.

Establishing Rules

A CSS stylesheet is analogous to a collection of styles in a word processor. Making a change in an element listed in a stylesheet changes all instances of that element in a document using that element. You might also choose to have dif-

ferent stylesheets for different page types to keep such structures as simple as possible. For example, you could have a paragraph style in both a copyright page and a body text page but format them differently by having the pages use a different stylesheet.

A stylesheet is a series of rules about formatting different text elements. For the convenience of humans – including you when you edit – the stylesheet usually adds an extra space between each rule. The basic format for a rule is:

```
SELECTOR {DECLARATION;}
```

The **SELECTOR** is the element to which the rule applies, and the **DECLARATION** is the formatting choice(s) that make(s) up the rule.

The selector can take several forms. The simplest is an HTML element, such as h2 (heading 2). It may or may not be followed by a class of the h2 tag, marked by a period. For instance,

```
h2 .example, h2, h3
```

would be a complete selector. You could add other selectors in a comma-separated list.

Alternatively, the selector can refer to a non-standard element named in the file that makes use of the stylesheet, with `<class="NAME">`. If the file uses `<div>`, then before the end of the section marked with `</div>`, you could also add `<id="NAME">`. You could also do the same with a section marked with a `` tag. Strictly speaking, a span tag is supposed to be used like a character style to mark a nested selection to be formatted differently, but in practice, the distinction between `<div>` and `` is not always observed. Since both function in the same way, the distinction is blurred.

The declaration follows the selector inside curly brackets. The declaration begins with the property, followed by the setting for the property, and ending with a semicolon. You could make a separate rule for each property, but to save space, you can place a semicolon-separated list of properties within the same curly brackets. So the complete rule might be:

```
h2 .example, h2, h3 {color:green;
weight:bold;font:Liberation Sans;}
```

If you want the stylesheet to be easily readable, you can take advantage of the fact that CSS ignores white space and separate properties by lines, so that the same example can be formatted as:

```
h2 .example, h2, h3
{
color:green;
weight:bold;
font:Liberation Sans;
}
```

No matter how you format the rule, the punctuation is simple but essential, so check it carefully.

CSS Positioning

Sometimes, CSS rules are used locally within a file where the formatting is needed. Because browsers generally can read a valid piece of CSS, this positioning is practical when a file has very few sections that require special formatting. However, this is like manual formatting in a word pro-

cessor: Both are cumbersome when you make changes while editing.

More often, CSS rules are used globally, where any changes only have to be made once. One choice is to place CSS in the `<head>` section of an HTML file (Listing 1). This location reduces the work and has the advantage of only requiring that you keep track of a single file.

However, increasingly, the preference is to place CSS in a separate file and to add a declaration at the top of a file. Assuming that the stylesheets and the files are in the same folder and the stylesheet is called "stylesheet", the declaration would be:

```
<link href="Styles.css"
rel="stylesheet" type="text/css"/>
```

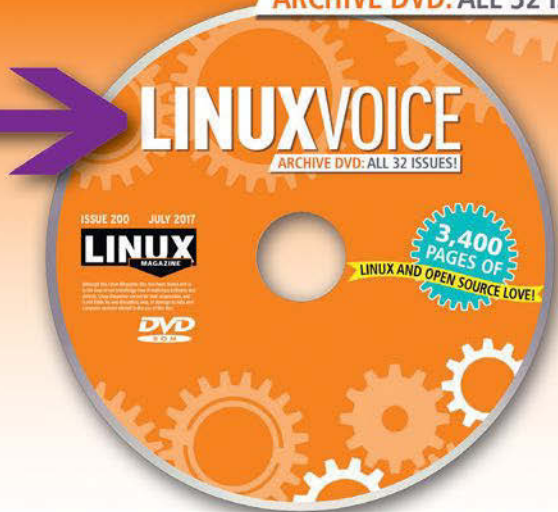
This link is useful for minimizing work and giving a uniform look to website pages or the different XHTML files in an EPUB file. If you want to change the look of a file or page completely, then all you need to do is replace the existing stylesheet with another of the same name, but with different rules. You never have to worry about accidentally changing content while editing formatting. The disadvantage is that you need to keep track of at least two files; the loss of the stylesheet means the loss of all your formatting.

Since both stylesheets in the head section and a separate file can be long, you might want to group properties into

Listing 1: CSS Placement

```
<html>
<head>
<style type="text/css">
h2 {color:black;font:Liberation Sans;}
h2 .example {color:green; weight:bold;font:Liberation Sans;}
p {color:black;font:Gentium;}
</style>
</head>
</body>
```

THE COMPLETE
LINUXVOICE
ARCHIVE DVD: ALL 32 ISSUES!



3,400 PAGES OF
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 EPUB, PDF, and HTML format!

Order now!

shop.linuxnewmedia.com

categories, as in the sample stylesheet in Listing 2. The `/*` marks the start of a comment and a `*/` the end. For convenience, you can add a color description designated in hexadecimal or RGB format, so you do not have to keep looking them up. Comments are ignored when a file is formatted using the stylesheet.

Basic CSS Properties

Most CSS uses depend heavily on formatting defaults. On web pages, you normally see fewer than two dozen CSS properties specified. However, for other purposes, such as ebook production, several dozen more are available. In fact, CSS has enough properties to reproduce the most common design features available in a word processor, although you should always check which features are supported by which browser and which ones your publisher permits. Most properties can take multiple values. You can check which properties are available on the CSS reference page [3], which not only lists valid values for each property but also links to

pages where you can experiment online before changing your own files.

As you work, remember that CSS supports nested properties. A passage marked by a listed property will automatically take the characteristics of the top-level property, unless there is a conflict. For instance, if a top-level property is `{color:red;}` and a nested property is `{font-style: italic;}`, the text that is italicized will be red like the surrounding text. Only if another color is specifically listed in the nested property will the italics use a different color. This property inheritance is the feature from which CSS derives its name: Properties cascade down from one declaration to others. Only when an end tag like `</p>` is used will a declaration cease to apply.

Another feature to note is that properties can overlap to an extent. For instance, `margin` and `padding` are different ways of setting the same feature (the space between the text and the edge of a page). Similarly, although you can use `margin` to set all the margins to the same value, you can set each `margin` individually with properties like `margin-left` and `margin-top`.

Most properties are one of two types: those that set appearance and those that set position and spacing. Among the most common properties for setting appearance are `font`, `font-size`, and `font-weight`. These properties give designers firm control over how text looks, but a case can be made for using them sparingly. For instance, if you are designing an ebook, being too specific about a font can cause display problems in some ereaders. Moreover, some ereaders allow readers to choose their own display fonts, so specifying too much about fonts is simply a waste of time. At the most, `serif` or `sans-serif` is all you might want to specify.

Other major text appearance properties are `color` and `background`. If you have any links, you might also want to choose a color for them using the properties `link`, `visited`, `hover`, and `active` – or, at least the first two, since users are unlikely to notice the last two distinctions. All these properties can take a basic ANSI color such as red or dark red, but,

if you prefer, you can use hexadecimal notation, such as `#00ff00` for blue, or an RGB color value such as `rgb(0,0,255)` for red. You can find hexadecimal and RGB color values in an app like Gimp.

Position-defining properties not only include `margin` and `padding`, but also, for images, `float-clear` (how text flows around an image), `height`, and `width`. Such properties can take values expressed as pixels, a percentage, or em spaces (i.e., the space take up by a lower-case letter “m”). As values, pixels are fixed, whereas percentages and em spaces are proportional.

Other position properties are available, but they should be used sparingly in some circumstances. For instance, multiple columns are generally avoided in many ebooks, as well as both numbered and bulleted lists, although CSS properties for both exist. Similarly, the general consensus is that `page-break-before` and `page-break-after` should be used only with an h1 heading.

These are only some of the most commonly used properties that can be specifically set. In many cases, the fewer you need to define, the less is likely to go wrong. Test as you work to avoid having to revise at the last moment before publication.

The Limits of CSS

CSS cannot reproduce all the features found in a word processor. If you have an elaborate layout, such as a brochure, you may be better off working in a word processor and then publishing in PDF, a format that can handle complex design. You can also add charts and diagrams as images. Even if it proves possible to reproduce a complex layout in CSS, the amount of effort and testing required is likely to be a burden.

CSS is designed for large bodies of text laid out in a regular column – in other words, articles and books. For such purposes, CSS provides the tools for a surprisingly sophisticated layout, especially when you take the time to study what is available. ■■■

Info

- [1] CSS: https://en.wikipedia.org/wiki/Cascading_Style_Sheets
- [2] W3C validation service: <https://jigsaw.w3.org/css-validator/>
- [3] CSS reference: <https://www.w3schools.com/cssref/>

Listing 2: A Basic Stylesheet

```
/* Character Set*/
@charset "utf-8";

/* Page Layout */
body {
margin: 0px;
}

/* Foreground (Black) & Background (White) */
html, body {
color: #000;
background: #fff;
}

/* Body Font */
body {
font: 1em sans-serif;
}

/* Heading Font and Size*/
h1, h2, {
font-family: sans-serif;
}
h1 { font-size: 2.5em; }
h2 { font-size: 2em; }

/* Link Colors: Blue and Dark Blue */
a:link { color: #00f; }
a:visited { color: #009; }
```

The sys admin's daily grind: Hue and Rasp Pi

More Light!

Since his Trådfri Smarthome article over a year ago, sys admin columnist Charly has been receiving messages from readers with two questions: “Can you do that with the Philips’ Hue system?” and “Can this also be done with a normal brightness sensor?” Yes and yes! *By Charly Kühnast*

In a previous issue [1], I described how I control a smart light in my living room with Ikea’s Trådfri system and Linux. The darker it gets, the more I turn up the lights. Around the same time, Martin Loschwitz then explained how he used a Raspberry Pi and a transmitter module to control Hue and other Zigbee devices [2].

Now I’m going to do it again, without the Zigbee module, but I expect to use a Hue bridge, which Martin’s approach doesn’t need, and because not everyone has a roof-top photovoltaic system, this time I focused on a simple brightness sensor.

Getting Started

First, I discovered the IP address of my Hue Bridge: 10.0.0.8. Then I pressed the button on the bridge and entered the following at the command line:

```
curl --data "{\"devicetype\": \"hue1r\"}" http://10.0.0.8/api
```

The hue1r string (for “Hue living room”) is freely selectable. I got some output like:

```
[{"success": {"username": "T3VGtGWmFUGJwHufxkudY1bizvqoZMhCSqulySwm"}]}
```

This unpronounceable string is an authentication string, which I have to specify with every call in the future. A (confusing) list of all connected devices is returned by the command:

```
curl --request GET 10.0.0.8/api/T3VGtGWmFUGJwHufxkudY1bizvqoZMhCSqulySwm
```

If you only need information about lights, you can append /lights to the command

or /lights/1, /lights/2, and so on for each lamp. The following command switches lamp 1 to medium brightness (value 125):

```
curl --request PUT --data "{\"bri\":125}" 10.0.0.8/api/T3VGtGWmFUGJwHufxkudY1bizvqoZMhCSqulySwm/lights/1/state
```

For the brightness sensor, I used a \$2.00 module named BH1750 (online, e.g., [3]). I connected this to the I2C bus of a Raspberry Pi (Figure 1). I used the Zero W model with WiFi – it has to communicate with the bridge.

Cat's Eyes

Raspbian comes with almost everything you need, with only a few packages to add:

```
sudo apt install build-essential wiringpi i2c-tools python-smbus
```

The program in Listing 1 reads the sensor in the program I named lux:

```
cc lux.c -lwiringPi
mv a.out lux && chmod 755 lux
```

If you run lux, you will see the current measured value for the photo sensor in lux units. Depending on this, you can now control your smart lighting. ■■■

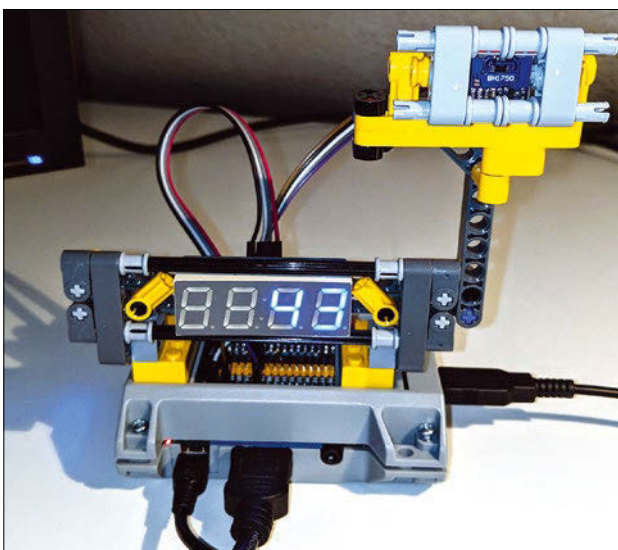


Figure 1: The case contains a Raspberry Pi Zero, with a seven-segment display peeping out. In the upper right corner of the picture, you can see the brightness sensor clamped in place.

Listing 1: lux.c

```
01 #include <wiringPiI2C.h>
02 #include <stdio.h>
03 int main (void) {
04     int handle = wiringPiI2CSetup(0x23);
05     wiringPiI2CWrite(handle, 0x10);
06     sleep(1);
07     int word=wiringPiI2CReadReg16(handle, 0x00);
08     int lux=((word & 0xff00)>>8) | ((word & 0x00ff)<<8);
09     printf("%d \n", lux);
10     return 0;
11 }
```

Info

- [1] “Charly’s Column: libcoap” by Charly Kühnast, *Linux Pro Magazine*, issue 202, September 2017, pg. 49: [http://www.linuxpromagazine.com/Issues/2017/202/Enlightened-libcoap/\(language\)/eng-US](http://www.linuxpromagazine.com/Issues/2017/202/Enlightened-libcoap/(language)/eng-US)
- [2] “Controlling Zigbee Devices with the Raspberry Pi” by Martin Loschwitz, *Linux-Magazin*, September 2017, pg. 76: <http://www.linux-magazin.de/ausgaben/2017/09/raspi-spricht-zigbee/> [in German]
- [3] Debo BH1750: <https://www.ebay.com/i/292393778070?chn=ps>

ADMIN Network & Security
Finding Phishing Domains Searching out sites that threaten your web presence
 ARIA TOSCA Build and test cloud service templates
 FREE DVD GParted
 KALI LINUX Patch the gaps before
 Manager Plus tool for managing the Director
 Gauntlit Security tool meets DevOps

ADMIN Network & Security
Securing Switch Ports Is your local network safe from internal intruders?
 Policy-Based DNS Adapting name resolution to your geography
 Yesod Web framework built for security
 Regexp in PowerShell
 Wind
 Ruby on Rails We study the latest version of the famous web framework
 CentOS7
 Tails 3.5
Real World AWS Building the cloud in your IT environment
 OpenShift What's new with Red Hat's container orchestration tool?
 Huginn Monitor and organize your work information
 for Managing words
 out this open SQL Server clone
AUTOMATING PEN TESTING

ADMIN Network & Security
Cloud Capacity Planning How much cloud do you really need?
 DOCKER SECURITY Why a container isn't as safe as a VM
 Best Practices for Securing Active Directory
 Jenkins Automated orchestration with continuous logging
 Jekyll Fast and lightweight HTML engine
 Policies

ADMIN Network & Security
CMS Shopping WordPress, Joomla, or Drupal? Choose the perfect tool for managing your web presence
 Packstack The easy way to roll out OpenStack
 Get Organized for a Vulnerability Assessment
 Terraform Configuration management for the DevOps age
 Is Your Website Secure? Use the Qualys server test to check for SSL problems
 Zodiac WX Inexpensive access point with OpenFlow support
 Protect Your Documents In the cloud with Azure Information Protection
 Windows In-Place Upgrade Continuous upgrades for Windows 10
 AWX Web-based console for Ansible

ADMIN Network & Security
Automating Pen Testing Patch the gaps before an slips inside
 Integration containers
 RE ANALYSIS malicious code finding infection
 communication in a
 IPv6 in Win Server 2016
 SQL Server 2017
 TLS Offload

6 issues per year!
GET IT FAST
 with a digital subscription!

shop.linuxnewmedia.com

REAL SOLUTIONS *for* REAL NETWORKS

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

It isn't all Windows anymore - and it isn't all Linux. A router is more than a router.

A storage device is more than a disk. And the potential intruder who is looking for a way around your security system might have some tricks that even you don't know. Keep your network tuned and ready for the challenges with the one magazine that is all for admins.

Each issue delivers technical solutions to the real-world problems you face every day. Learn the latest techniques for better:

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

on Windows, Linux, and popular varieties of Unix.

SUBSCRIBE NOW
SAVE 30%

ADMIN
Network & Security

shop.linuxnewmedia.com

A Go terminal UI for displaying network adapters in real time

Classics Repackaged

Even command-line lovers appreciate a classic terminal UI. Mike Schilli shows how to whip up a Go program that dynamically displays network interfaces and IPs. *By Mike Schilli*

Every time I connect my laptop to a router for diagnostic purposes, the question arises: On which dynamically assigned IP address will the router see the laptop? After all, you need to enter a router address on the same subnet to display the router's admin page (Figure 1).

To do this, I used to type `ifconfig` several times in a terminal window and ex-

tracted the desired address from the mess of data printed next to:

```
inet 192.168.1.1 netmask 0xffffffff
```

I thought there must be an easier way. How about a program that figures out all available network interfaces every couple of seconds, sorts them into a list, and dynamically displays their IP addresses? In a graphical user interface (GUI) that popped up, the relieved user could then watch a plugged in USB adapter appear as a new network interface and see the IP assigned to it by DHCP as well.

But, it doesn't have to be a genuine graphical application, as I recently presented in this column, elegantly programmed with GitHub's Electron framework [1]. Command-line friends prefer terminal UIs à la top, instead; they can be started, read, and closed quickly, without keyboard addicts having to leave the terminal window at all or reach for the unloved mouse.

Ready To Go

What the `ifconfig` command-line tool prints is something that the `net` package in Go already has up its sleeve, and it provides network adapter configuration in the form of a data

structure. Listing 1 [2] shows the implementation of an `ifconfig` helper package that exports an `AsStrings()` function, which returns a formatted list of all network interfaces detected on the current machine.

In line 13 of Listing 1, the `Interfaces()` method from the Go `net` package returns a series of network interface structures, through which the `for` loop iterates with the help of `range` as of line 14. The `range()` function for the delivered slice (a dynamic window on a static array in Go) not only returns the current element for each loop iteration, but also its index into the slice, which is not needed here and is therefore assigned to the `_` pseudo-variable and thrown away.

The string formatter in line 15 sets the network variable to the name of the interface (e.g., `eth0` for the first Ethernet adapter found), right-justified with a maximum length of 10 characters. The IP addresses that the interface listens on are retrieved by the `AddrS()` function.

Common in Go, the function returns two parameters, first a slice with all discovered IPs and then an error variable, hopefully set to `nil`, indicating that everything went fine. To save space in this article, that second error variable is set to `_` in line 17 of Listing 1, thus discard-

Author

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



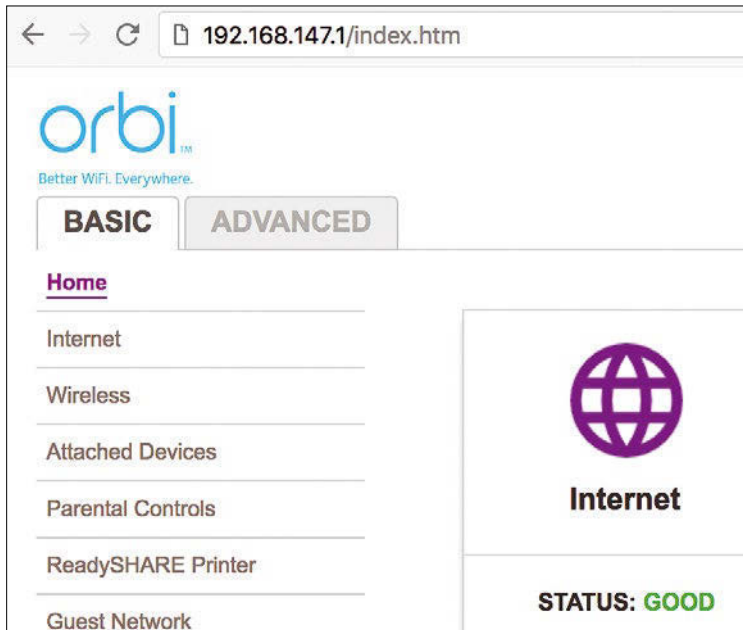


Figure 1: This machine uses an IP address on subnet 192.168.147.1/24 to connect to the router.

ing errors – something you should not do on a production system.

If the device does not have an IP assigned to it, the discovered network interface is not relevant and line 19 uses `continue` to jump to the next one. Of potentially multiple IPs per interface, only the first one is of interest on my simply structured laptop. Since the network there may be in CIDR format instead of an IP (e.g., 192.168.1.1/24), the `Split()` function from the `strings` package splits off the netmask in line 21 so that the `addr` variable contains only the actual IP as a string.

Because I still work with good old IPv4 at home, line 24 blocks IPv6 addresses. The call to

```
net.ParseIP(addr).To4()
```

tries to convert any addresses discovered to IPv4 format, which only works for IPv4 addresses and returns an error value other than `nil` for IPv6 addresses. If your home setup is up-to-date and uses IPv6, this filter condition needs to go, of course, and you'll see IPv6 addresses in the display as well.

Line 29 sorts the formatted list alphabetically before the return statement in the following line returns it to the caller.

Compiler Playing Dumb

When you are picking names for new functions in Go, remember that in a

package like `ifconfig`, functions starting with a lowercase letter are not exported. If the importing main program called an `as_strings()` function implemented in the package, the Go compiler would refuse to comply and simply claim that such a function does not exist. Instead, the function in `ifconfig` must begin with an uppercase letter: The capitalized `AsStrings()` will later also be found by the main program importing the package.

Go compiles everything that belongs to a program into a static binary. For the compiler to find the imported package in Listing 1 when the main program is put together, it must find the static *.a library generated for it in the Go path (`$GOPATH`), which is typically found below `~/go` in your home directory. If the library goes by the name of `ifconfig`, its source code must be stored in a newly created directory named `ifconfig` below `src` and be installed from there with `go install`:

```
dir=~/.go/src/ifconfig
mkdir -p $dir
cp ifconfig.go $dir
cd $dir
go install
```

Listing 1: ifconfig.go

```
01 package ifconfig
02
03 import (
04     "fmt"
05     "net"
06     "sort"
07     "strings"
08 )
09
10 func AsStrings() []string {
11     var list []string
12
13     ifaces, _ := net.Interfaces()
14     for _, iface := range ifaces {
15         network := fmt.Sprintf("%10s",
16             iface.Name)
17         addrs, _ := iface.Addrs()
18         if len(addrs) == 0 {
19             continue
20         }
21         split := strings.Split(
22             addrs[0].String(), "/")
23         addr := split[0]
24         if net.ParseIP(addr).To4() != nil {
25             network += " " + addr
26             list = append(list, network)
27         }
28     }
29     sort.Strings(list)
30     return list
31 }
```

This command sequence creates the static library `ifconfig.a` below `pkg/linux_amd64` in the Go path; later, when building the main program, the Go compiler links the library statically with it.

The `termui` project on GitHub [3] is used as the terminal GUI for the utility. The beauty of Go is that its code can be installed directly from the web using the `go get` command-line tool:

```
go get -u github.com/gizak/termui
```

The `get` command fetches it from GitHub, compiles it, and installs the libraries created by this step in the Go path, where the compiler will find them later, if a Go program demands they should be linked with it. The `-u` flag tells `go get` not only to install the required

package, but also to update any dependent packages.

Exciting Events

Like most GUIs, termui is event-based. The user initially defines some widgets, such as list or text boxes, arranges them with a layout tool in 2D space, starts the loop, and then intercepts events such as *Terminal window size reduced* or *Key combination Ctrl + C pressed* or *The timer that starts every second has just elapsed*. For today's network tool, Listing 2 defines two different widgets, as shown in the screenshot in Figure 2: a list box at the top, which lists the available network interfaces with their IPs as entries, and a text box at the bottom, which only reminds the user to press the *q* key to exit the program.

After line 4 has imported the termui package, assigning it the *t* abbreviation,

the main program calls `Init()` to initialize the GUI for the termui package, wiping the terminal window clean and setting it to graphics mode. At the end of the main program, the `Close()` call reverts these actions, and a normal text terminal is restored. Thanks to the `defer` keyword, which comes as part of the Go standard feature set, the cleanup is planned in line 18, but Go delays action until leaving the `main` function.

Practical Box

The layout algorithm in termui arranges the widgets passed to it in a 12x12 grid. The `AddRows()` function in line 33 accepts layout rows created with `NewRow()` as arguments; their arguments in turn are columns created with `NewCol()`. The latter function takes the width of the column as the first argument; in Listing 2, this means all 12 of a total of 12 grid

squares. The second parameter is an offset as a spacer, which remains unused here at 0.

The `Align()` function called in line 40 builds the widget grid internally; the following `Render()` puts the whole thing on the screen. Now the only thing left is to intercept any events that occur – for example, if the user increases or decreases the size of the terminal window. In this case, the raster engine kicks in thanks to the `/sys/wnd/resize` handler in line 45. The code first uses `TermWidth()` to determine the new terminal width and then calls the `Align()` function to distribute the widgets across the available space once again. The following call to `Render()` refreshes the display in a smooth, single action.

Now the list box with the displayed network interfaces is originally empty, because `Items` – a “field” (attribute) type

Listing 2: iftop.go

```

01 package main
02
03 import (
04     t "github.com/gizak/termui"
05     "ifconfig"
06     "log"
07 )
08
09 var listItems = []string{}
10
11 func main() {
12     err := t.Init()
13     if err != nil {
14         log.Fatalln("Termui init failed")
15     }
16
17     // Cleanup UI on exit
18     defer t.Close()
19
20     // Listbox displaying interfaces
21     lb := t.NewList()
22     lb.Height = 10
23     lb.BorderLabel = "Networks"
24     lb.BorderFg = t.ColorGreen
25     lb.ItemFgColor = t.ColorBlack
26
27     // Textbox
28     txt := t.NewPar("Type 'q' to quit.")
29     txt.Height = 3
30     txt.BorderFg = t.ColorGreen
31     txt.TextFgColor = t.ColorBlack
32
33     t.Body.AddRows(
34         t.NewRow(
35             t.NewCol(12, 0, lb)),
36         t.NewRow(
37             t.NewCol(12, 0, txt)))
38
39     // Initial rendering
40     t.Body.Align()
41     t.Render(t.Body)
42
43     // Resize widgets when term window
44     // gets resized
45     t.Handle("/sys/wnd/resize",
46         func(t.Event) {
47             t.Body.Width = t.TermWidth()
48             t.Body.Align()
49             t.Render(t.Body)
50         })
51
52     // Refresh every second
53     t.Handle("/timer/1s", func(t.Event) {
54         lb.Items = ifconfig.AsStrings()
55         t.Render(t.Body)
56     })
57
58     // Keyboard input
59     t.Handle("/sys/kbd/C-c", func(t.Event) {
60         t.StopLoop()
61     })
62     t.Handle("/sys/kbd/q", func(t.Event) {
63         t.StopLoop()
64     })
65
66     t.Loop()
67 }

```

in Go, as part of a struct (mixed data structure) – wasn't initialized in the beginning. Line 54 now takes care of this with the handler called once per second for the `/timer/1s` event. It calls the `AsStrings()` function from the `ifconfig` package previously shown in Listing 1 and assigns to the list box a pre-formatted slice of strings with the network adapters and their IPs. For the refreshed list to appear on the screen, the graphic

manager's `Render()` function must be called again.

All's Well That Ends Well

To allow the user to exit the program gracefully without the terminal getting stuck in graphics mode and becoming unusable, the handlers `/sys/kbd/C-c` and `/sys/kbd/q` in lines 59 and 62 intercept the `Ctrl+C` and `Q` keystrokes and use `StopLoop()` to stop the event

loop, which in turn lets the blocking `Loop()` function in line 66 return before `main()` finishes.

But wait, before that happens, the `defer t.Close()` command, which was called previously in line 18, quickly calls the graphic manager's cleanup function, which kindly cleans up everything to leave the terminal as it was when the program started.

You can compile the program by typing:

```
$ go build iftop.go
```

If the `ifconfig` package from Listing 1 is correctly installed, all code will be wrapped up in a binary that clocks in at about 3MB. It contains everything, so you can just copy it to another machine with a similar operating system and launch it there if so desired.

When you finally launch the program by typing `./iftop` at the command line, you will see a tidy screen with the elements shown in Figure 2: the initially empty list box at the top, and the text field at the bottom. After one second, when the timer expires for the first time, a list of the network interfaces with their IP addresses appears in the list box.

If you want to test the dynamic behavior, you can now plug a network cable into your laptop (e.g., via USB), connect it to your router, and see a new entry appear in the list box almost immediately. Conversely, an existing entry should disappear as soon as a network interface goes away (e.g., when the laptop's wireless signal is turned off).

The `termui` project offers a generous helping of eye candy in the terminal, from progress bars to bar and pie charts (Figure 3), and `termui` is not the only terminal UI framework in Go: `gocui`, `clui`, `uui`, `tui-go`, and others also vie for the developer's favor. A blog post online [4] provides an overview of the advantages and disadvantages of the individual releases. It would be quite possible to build a sophisticated terminal application like `lazygit` [5] with one of these frameworks or a kind of Norton Commander (also known as `Midnight Commander` on Linux) for retro-looking file manipulation – state-of-the-art again 35 years after MS-DOS! ■■■

Info

- [1] "Using the Electron Framework to Weed Out Images" by Mike Schilli, *Linux Pro Magazine*, issue 216, November 218, p. 46.
- [2] Listings for this article: [ftp://ftp.linux-magazine.com/pub/listings/linux-magazine.com/218/](http://ftp.linux-magazine.com/pub/listings/linux-magazine.com/218/)
- [3] `termui` on GitHub: <https://github.com/gizak/termui>
- [4] "Text-Based User Interfaces" by Christoph Berger: <https://appliedgo.net/tui/>
- [5] `lazygit`: <https://github.com/jesseduffield/lazygit>

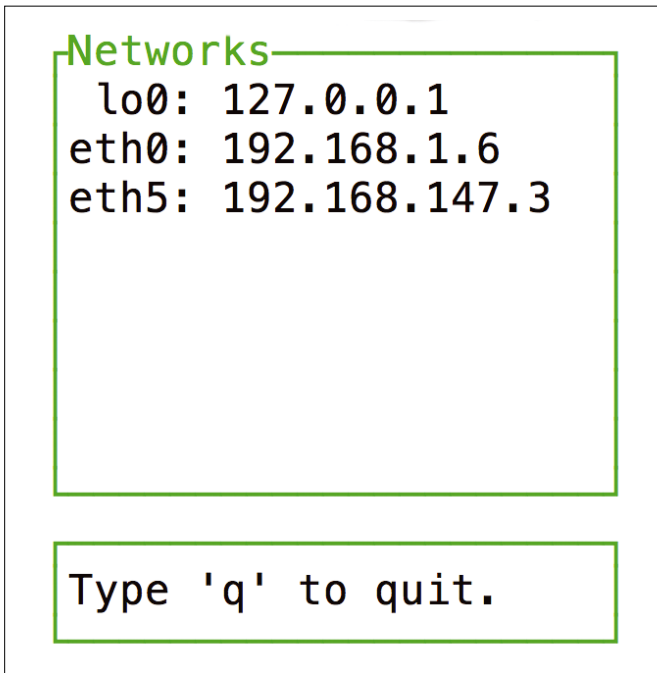


Figure 2: The Go program displays a dynamic list of network ports.

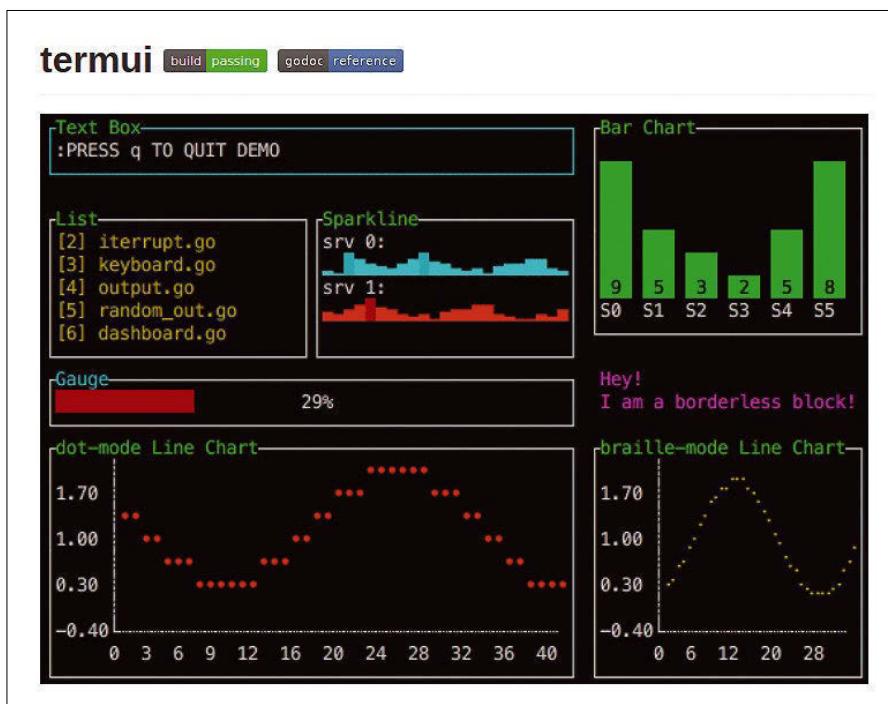


Figure 3: The `termui` project's Demo dashboard shows what's possible.



Facebook releases its own OOM implementation

Contract Killer

When a Linux system runs out of memory, a special agent, the out-of-memory killer, rushes to its aid. Facebook has now introduced its own OOM killer. What makes it different from its kernel-based counterpart? And what is an OOM killer really? *By Martin Loschwitz*

If you have not placed an order for a large server for a long time, you will probably rub your eyes in amazement the next time you order a new device: Configurations with terabytes instead of gigabytes of RAM are easy to

get, and you don't need to be a millionaire to buy them. Gone are the days when people were proud of every single gigabyte (Figure 1).

Some buyers don't even worry about RAM anymore and just assume the system

will have enough; however, this might be a little too optimistic, even on a modern system. Servers still sometimes come up short on RAM, and when they do, it can have dramatic consequences: If a component such as systemd needs RAM and cannot allocate it, the system will malfunction or stop working. To avoid a RAM shortage bringing computers to their knees, the Linux kernel has a watchdog on board: the out-of-memory killer, or OOM killer for short. In an emergency, OOM frees up memory by shooting down processes in a targeted way; the memory is then available for other, presumably more important purposes.

Many legends and horror stories are centered on the OOM-killer, and the admin's sense of humor is typically strained when they see kernel messages in the log saying that the killer has struck again (Figure 2). The reason for the anxiety is that it is large applications,

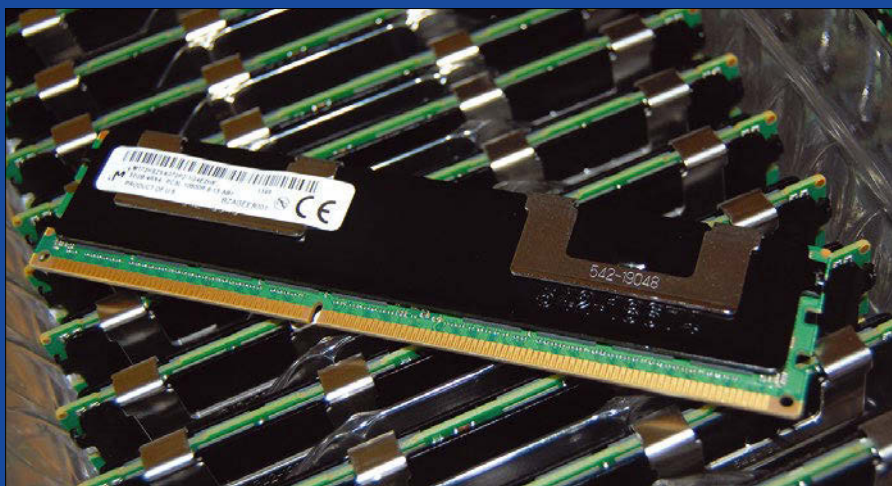


Figure 1: RAM is no longer a scarce resource, but no system is immune to an OOM situation.


```
[ 44.864054] Kernel panic - not syncing: Out of memory: system-wide panic_on_oom is enabled
[ 44.864054]
[ 44.864183] Pid: 3656, comm: mysqld Tainted: G      0 3.2.0-4-amd64 #1 Debian 3.2.41-2
[ 44.864265] Call Trace:
[ 44.864309] [] ? panic+0x95/0x1a2
[ 44.864364] [] ? check_panic_on_oom.part.3+0x59/0x59
[ 44.864428] [] ? out_of_memory+0x221/0x337
[ 44.864488] [] ? __alloc_pages_nodemask+0x629/0x7aa
[ 44.864552] [] ? alloc_pages_current+0xc7/0xe4
[ 44.864612] [] ? filemap_fault+0x24f/0x33e
[ 44.864690] [] ? __do_fault+0xc8/0x3ac
[ 44.864748] [] ? handle_pte_fault+0x298/0x79f
[ 44.864808] [] ? pte_offset_kernel+0x16/0x35
[ 44.864879] [] ? do_page_fault+0x312/0x337
[ 44.864950] [] ? aio_queue_work+0x2c/0x2c
[ 44.865008] [] ? page_fault+0x25/0x30
[ 44.865080] Rebooting in 10 seconds...
```

Figure 2: The OOM killer in the Linux kernel either kills individual processes or reboots the server if the kernel is configured to do so.

such as Java, that the OOM killer targets as its victims.

Java is not famed for being very sparing with resources, but it is usually necessary for running the application for which the server exists. If the OOM killer shoots down Java on a Tomcat system, a load balancer usually catches the problem, but the server taken out in this way is still gone at the end of the day.

This article introduces the current OOM implementation in Linux and explains how it works. I will then compare this standard implementation with an alternative approach chosen by Facebook.

How OOM Situations Occur

Even servers with huge amounts of RAM can get into situations where the available system RAM is not sufficient. This is because the Linux kernel uses certain ways and means to allocate memory as efficiently as possible.

If you have ever called top and looked at the RAM statistics, you will be aware that even on systems with a large amount of RAM and very little load, the display for RAM utilization is often close to the 100 percent limit, even if the system has nothing to do (Figure 3).

The Linux kernel is the interface between the hardware on one side and the programs on the other. If a program wants memory, it asks the kernel for using a system call

like malloc(). However, it takes too long for the kernel to first search for free memory and then make the requested amount available.

Instead, the kernel preempts: It divides the entire available memory into segments, known as memory pages. In addition, the kernel remembers which pages are already assigned to the running programs and which are thus still available. If a program now comes along and uses RAM, the kernel simply assigns it a memory page from the list of free pages. Because the memory pages are not all the same size, the kernel also has a certain degree of flexibility and can ensure that there is not too much waste.

Waste Is Bad

It is important to avoid waste to the greatest extent possible. Even if you have an arbitrary amount of RAM at

your disposal, you will still want to use it as well and efficiently as possible. For many years, the Linux kernel has supported a function that many admins consider equivalent to opening up the proverbial Pandora's box – overbooking RAM.

Roughly speaking, it works like this: The kernel assigns memory pages to requesting programs as usual, but more in total than would actually be available through the physically available working memory. This does not directly cause OOM problems – they are caused by programs that require too much RAM.

However, RAM overcommitment increases the risk of OOM situations because the kernel does not rigorously deal with potential difficulties in advance. If Linux did not allow applications to allocate more memory than actually exists, crashes due to a lack of memory would be unthinkable because applications would simply see an error message when they tried to claim more memory than available.

The Linux approach is different. The kernel speculates that allocated memory will never be fully used. The vm.overcommit_memory=sysctl variable manages everything else: If it is set to 0, which is the default value, the kernel uses a heuristic approach to calculate how much RAM is actually free. It then sets this in relation to the memory that a requesting application wants to have. If the calculations are positive,

```
 1  [|||||] 16.4%] Tasks: 80, 168 thr; 2 running
 2  [||] 4.0%] Load average: 1.17 2.31 2.42
 3  [||] 2.7%] Uptime: 1 day, 05:11:34
Mem[|||||]1007/7479MB]
Swp[|]0/8099MB]

  PID USER      PRI  NI  VIRT   RES   SHR  S  CPU% MEM%   TIME+  Command
16499 joe        20   0 1904M 325M 73396 S 13.7  4.4  3:26.71 cinnamon --replace
16157 root       20   0  305M 74336 37616 S  3.9  1.0 16:52.70 /usr/bin/X :0 -audi
18419 joe        20   0 25784  3516  2888 R  1.3  0.0  0:00.35 htop
17369 joe        20   0  581M 27724 21680 S  1.3  0.4  0:13.16 gnome-terminal
17391 joe        20   0 1363M 291M 47740 S  0.7  3.9  1h11:10 simplescreenrecorde
18077 joe        20   0 1363M 291M 47740 S  0.7  3.9  2:12.44 simplescreenrecorde
18079 joe        20   0 1363M 291M 47740 S  0.7  3.9  0:02.26 simplescreenrecorde
18421 joe        20   0  443M 22788 18848 S  0.0  0.3  0:00.25 gnome-screenshot --
18080 joe        20   0 1363M 291M 47740 S  0.0  3.9  0:05.40 simplescreenrecorde
16501 joe        20   0 1904M 325M 73396 S  0.0  4.4  0:07.82 cinnamon --replace
18078 joe        20   0 1363M 291M 47740 S  0.0  3.9  0:39.42 simplescreenrecorde
16531 joe        20   0 1904M 325M 73396 S  0.0  4.4  0:00.09 cinnamon --replace
16439 joe        20   0  886M 40912 32292 S  0.0  0.5  0:01.20 /usr/lib/x86_64-lin
  1 root       20   0 33884  4244  2552 S  0.0  0.1  0:01.79 /sbin/init
F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit
```

Figure 3: Even on systems with a generous helping of RAM and little load, it often looks like all the RAM is allocated – this is due to the kernel's memory paging.

the program gets the memory, even if the amount of allocated memory becomes larger than the actual memory available in the system.

`vm.overcommit_memory=1` makes the kernel even more radical: In this case, the kernel skips the heuristic analysis and approves every request for RAM. But if you set the value to 2, RAM overbooking is switched off.

What Really Helps

If you think that it is sufficient to deactivate RAM overbooking on the basis of the previous explanations, you are wrong. The OOM problem is not caused by overbooking RAM, but by programs that continuously allocate too much RAM. And unfortunately, they usually do this unpredictably and for a variety of reasons. Often the root of the problem is simply a programming error, which causes the affected program to overburden the RAM. Occasionally, it actually happens that a system needs more RAM than is available to process incoming requests.

If you are confronted with OOM situations, you should first try very carefully to find the cause. If the emergency is not based on a programming error and the OOM situations occur regularly and reproducibly, the long-term solution can only be more hardware. You can either put more RAM into the affected servers or scale the setup horizontally.

If you are dealing with a programming error, it is a good idea to find it and repair it – in collaboration with the developers if necessary. Troubleshooting in such cases can be tough and time consuming. But if OOM problems occur after an update where there were none before, a bug is most likely the trigger.

The Unpopular OOM killer

The kernel's OOM killer definitely is only a helper. However, it fulfills an important role if your primary goal is to prevent the system from crashing: In case of an emergency, it kills programs that would otherwise – with a high probability – cause the system to crash due to RAM over-allocation. Put simply, the task of the OOM killer is to give the admin some time to deal with the actual problem in detail, without the entire environment blowing up in the admin's face.

As a helper construct, the kernel's internal OOM killer often does more evil than good – if you believe Facebook. I'll explain why the social media giant has been critical of the kernel OOM solution, but first I need to explain a little more about how the current OOM killer works in Linux.

Holistics Instead of Heuristics

The way the OOM killer identifies its potential victims on the running system has changed dramatically in recent years. The first OOM implementation in Linux, which was in use for many years, was essentially based on heuristics: Using many parameters, the kernel tried to find out which programs are unimportant for providing the elementary functions and can be sent to the happy hunting grounds without causing too much fuss.

The function that is responsible for this in the kernel memory management is appropriately known as `badness()` – it calculates which process on the running system has the highest level of badness, and thus generates a list in descending order.

However, the heuristics used were not particularly comprehensible, and so it was Google – in the person of David Rientjes – who established a completely new implementation of the OOM killer in Linux in 2010. The function that is responsible for identifying the problematic processes is still known as `badness`, but beyond that, not much has remained the same.

The `badness` function now follows a rather simple approach: It is interested almost exclusively in the memory consumption of individual processes, throwing all of the system's processes onto the scales, and then posing the question as to how it can free up as

much memory as possible by switching off as few processes as possible.

For each process, the kernel calculates the OOM score, which, by the way, can also be read using `cat` from the `/proc` filesystem (`/proc/PID/oom_score`). If an OOM situation arises, the OOM killer starts to terminate the processes with the highest OOM scores one after another.

How exactly the OOM score is calculated is much easier to answer since the patches by David Rientjes were introduced: For each process, the kernel evaluates how much memory it actually uses. Memory is defined as the sum of the working memory and the swap memory, although swap memory is becoming increasingly rare on today's systems. The rule of thumb is: The more memory a process uses at the time of the OOM rating, the higher it is listed on the kill list.

Exceptions Confirm the Rule

David Rientjes has implemented a few exceptions to avoid annoying effects. Basically, the kernel always subtracts 30 points from the OOM score for root processes because the system administrator's processes are potentially more important than those in user space. In addition, the admin can also influence the score: For this purpose, there is an `oom_score_adj` file in the `/proc` filesystem that you can use to adjust the value for each process. Values from `-1000` to `+1000` are possible.

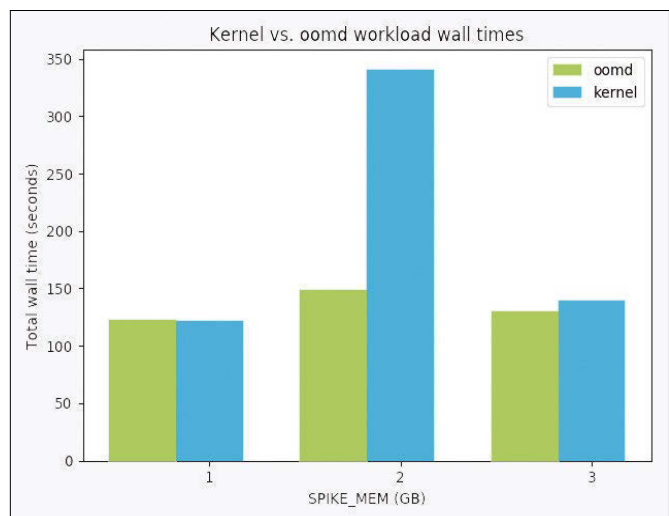


Figure 4: Maximum time saving: The OOM killer holds a system in a stranglehold for almost six minutes in an OOM scenario; OOMd only needs about three minutes.

If the admin gives a process `-1000`, the OOM killer will not kill this process. However, setting the value to `+1000` is like painting a large X on the process's belly; it increases the probability that this process will be the first to die an OOM death.

Facebook Not Amused

Compared to the apparent erratic way the `badness()` function worked until 2010, the current OOM implementation is easy to understand and sensibly designed. But not all companies that use Linux on a large scale see it that way.

One very prominent critic of the current OOM implementation is Facebook: The company is known to rely on Linux in its data centers, albeit in a heavily modified form.

The Facebook developers summarized their problems with the kernel's own OOM killer on the net, stating that the OOM killer reacted unpredictably. The possibilities for influencing its function were too limited, and the configuration options inadequate. And be-

cause the OOM killer resided in the kernel, it was also extremely sluggish: There can be a pretty long wait between the time when the kernel realizes that there may be a memory problem, and the time when it actually does something about it and releases memory (Figure 4).

The kernel kills the processes one after the other, re-evaluates the situation, and has to empty the page cache in the meantime before looking at the next process, so the cycle is continually restarted until the system is running stably once again.

In the worst case, according to Facebook, livelock situations lasting more than 30 minutes can occur. As a reminder: A deadlock is a situation in which all components of a system are waiting for another component on the same system. Livelocks work in the same way, but the relations between the waiting components change at regular intervals. While the kernel could currently be waiting for the remains of a program to disappear from the page cache, it might be waiting for the OOM

score to be calculated for a certain process a short while later.

Facebook's annoyance with this situation is understandable: During the 30 minutes that such a scenario allegedly takes, the system is practically unusable. Only a hard reboot is an effective measure, but that is exactly what Facebook wants to prevent.

OOM as a Holistic Measure

OOMd [1], which Facebook presented in August 2018, cuts off old braids and certainly takes a very bold approach to the subject of OOM. The biggest difference with the existing OOM implementation in Linux is probably the fact that Facebook's OOM killer does not run in kernel space, but as a normal application in user space. This makes OOMd a real novelty, as OOM has undisputedly been part of the kernel domain until now.

OOMd doesn't have big technical disadvantages – ultimately, if it is running with `sys admin` privileges, it can kill a process simply by issuing a `SIGKILL`. The OOM killer in the kernel doesn't do

The intersection of DEVELOPMENT and OPERATIONS

Check out our new ADMIN DevOps corner!

www.admin-magazine.com/DevOps

SPONSORED BY



Linux
Professional
Institute

anything wildly different at the end of the day.

In return, the advantages of the user-space implementation are obvious: In particular, OOMd is far more flexible than the kernel implementation could ever be.

Adding PSI

If you want to try out OOMd right after reading this article, be warned: You still need the kernel. The functions do not reside in the kernel in OOMd's case. But OOMd in user space depends on receiving as much information as possible about the current state of the system from the kernel. A suitable interface already exists in the form of Linux PSI [2], a component that reports on kernel memory, CPU, and IO pressure metrics. Linux PSI is not yet in the kernel. If you want to use OOMd, you have to build PSI into your kernel, a task that is made easier because PSI is available as a kernel module.

Always on Your Guard

OOMd is also modular. In addition to PSI, the service relies on cgroups v2 under-the-hood; this is a kernel-based function for limiting and assigning resources and permissions. OOMd does not force the implementation of cgroups but uses cgroups v2 only as a second data source for its own monitoring of the system with a focus on RAM.

The dominant principle behind OOMd is that caution is far better than indulgence. Monitoring plays an important role: Unlike the kernel implementation of the memory killer, OOMd is designed to take action before a system-wide OOM situation occurs. To this end, the service continuously scans its main data sources – cgroups and PSI – and, in the event of an issue, automatically takes action before a genuine OOM situation actually occurs (i.e., before a `malloc()` call fails).

Because OOMd resides in user space and not in the kernel space, it benefits from far greater flexibility, especially with regard to extensions. It is thus not surprising that Facebook has designed the OOMd with a plugin interface that can be used to load external functionality. The plugins are directly integrated into the OOMd workflows – but the only plugin that comes with the OOMd is the one for killing processes. However, many

```

"cggroups": [
  {
    "target": "system.slice",
    "kill_list": [
      {"chef.service": {"kill_pressure": "60", "max_usage": "100"}},
      {"sshd.service": {"max_usage": "inf"}}
    ],
    "oomdetector": "default",
    "oomkiller": "fbje"
  },
  {
    "target": "workload.slice",
    "kill_list": [],
    "oomdetector": "default",
    "oomkiller": "default"
  }
],
"version": "0.2.0"

```

Figure 5: The OOMd configuration file is neatly organized and offers levers and switches for all important parameters in JSON format.

options are conceivable, such as the option of performing various other tasks on a system before killing a process.

Simple Configuration

OOMd is written in C++ and uses the JSON notation as its configuration format. The results are very clear: The cgroups directive allows different procedures to be defined for individual cgroups, where `target` stands for the target group to which the entry refers. The other entries in each cgroup directive determine, among other things, which handlers and actions are used for OOM events. Figure 5 provides an example that makes it quite clear how flexible OOMd is in practice.

The example specifies that, when an OOM situation occurs, the `chef` service should be stopped first if it accounts for more than 60 percent of the total RAM load or has allocated more than 100MB of RAM for itself. The developers, however, have given the OpenSSH daemon (`sshd`) a survival guarantee; the `max_usage: inf` entry specifies that OOMd must never shut down the service under any circumstances.

This is also very useful, because without `sshd`, only a VNC or KVM console would remain for the remote login, and in the case of the kernel OOM, it is quite conceivable that the `sshd` would get into the firing line as one of many memory users. If you use OOMd, on the other hand, you can completely rule this out.

The Facebook developers heap praises on their own solution, by the way. In the

documentation on GitLab, they state that the new OOMd has completely eliminated the previously described livelock situation that resulted in waits of 30 minutes or more.

Conclusions

Facebook brings an urgently needed and pleasantly fresh breeze into the OOM debate with its in-house solution: It now becomes clear that OOM is not a task that the kernel must handle. And it is also clear that functionality and flexibility can be achieved in user space that would be unthinkable in kernel space. The userspace OOM killer published by Facebook is interesting in this respect and will hopefully see rapid development.

And that additional development necessary: Currently, OOMd's functionality is limited to little more than what the existing OOM killer in the kernel can do. Nevertheless, Facebook explains that the tool works far better in everyday life than the original Linux solution. It is therefore almost impossible to predict what will happen if OOMd attracts a larger fan base and is available in packaged form for the usual distributions. It's quite possible that Facebook has succeeded in a real coup. One thing is certain: If you are looking for a virtual contract killer, you now have a choice. ■■■

Info

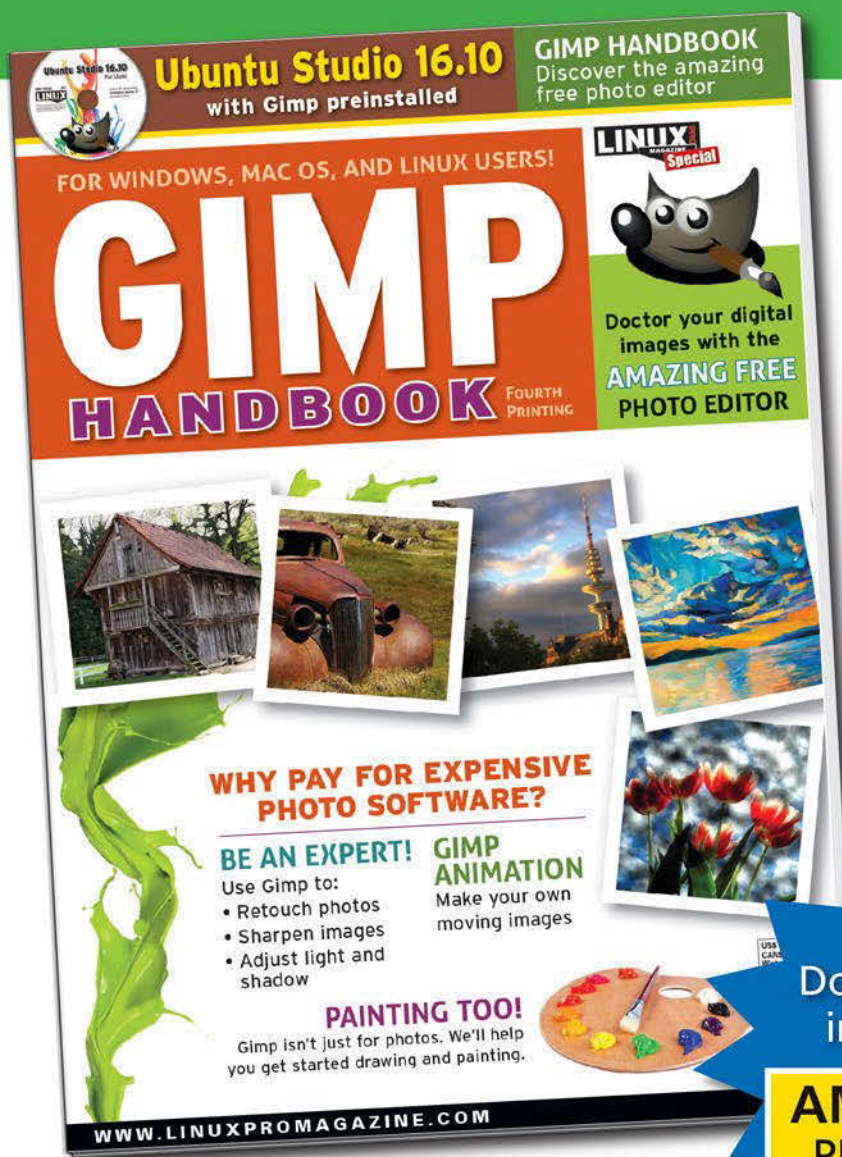
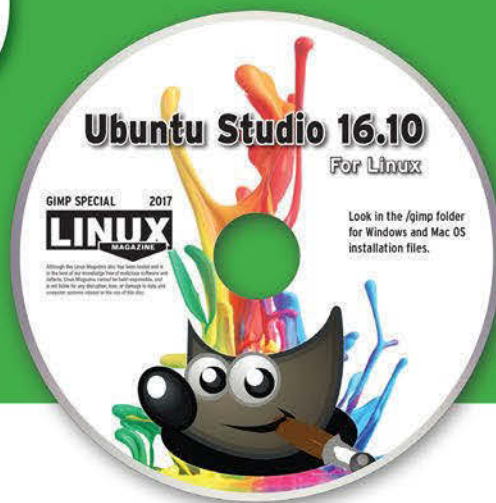
[1] OOMd: <https://github.com/facebookincubator/oomd>

[2] Linux PSI: <http://git.cmpxchg.org/cgit.cgi/linux-psi.git/>

Shop the Shop

shop.linuxnewmedia.com

GIMP HANDBOOK



**SURE YOU
KNOW LINUX...**
but do you know **Gimp?**

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

Order now and become an expert in one of the most important and practical open source tools!

Gimp
Doctor your digital images with the
AMAZING FREE PHOTO EDITOR!

Order online:
shop.linuxnewmedia.com/specials



FOR WINDOWS, MAC OS, AND LINUX USERS!



Password-free authentication: FIDO2 and WebAuthn

Trustworthy

FIDO2 authentication with WebAuthn may be sounding the end of the password age. *By Markus Feilner*

Fido was a loyal soul – the name literally means faithful, loyal, trustworthy [1]. The mongrel found its way to the Lincoln household and quickly became a family member. Dirty paws, sleeping on

the sofa – the dog was allowed to do everything (Figure 1). He even shared his master’s fate: Only a few months after the fatal assassination of US President Abraham Lincoln, a drunk stabbed him [2].



Figure 1: Loyal to the death: Abraham Lincoln’s dog Fido.

In 2012 a new player, the FIDO (Fast Identity Online) Alliance [3] – not to be confused with Fidonet [4], a bulletin board system (BBS) from the 1980s and 1990s that older readers will remember as popular with hackers and geeks – came onto the market (see the “The FIDO Alliance” box).

FIDOsofpy

According to the FIDO Alliance website, their goal is to combine transaction-secure, strong authentication with good

The FIDO Alliance

The beginnings of the non-profit FIDO Alliance date back to 2009, with roots in biometrics and PayPal environments. Since its founding in July 2012, more than 250 industry representatives have gathered under its roof [5], including financial institutions, computer hardware and processor manufacturers, software and information companies, security organizations, and, since October 2015, the German Federal Office for Information Security. In 2013, work began on a passwordless authentication protocol.

According to the FIDO Alliance, 3.5 billion user accounts worldwide use FIDO authentication, 80 percent of available mobile devices support the process, and more than 400 FIDO-certified devices are available.

Lead Image © Studio Porto Sabbia, 123RF.com

usability while preventing fraud and providing the provider with the most efficient and uniform authentication mechanism possible.

The approach aims to combine biometrics and two-factor authentication, which is achieved when a user deposits a key on a server and then confirms subsequent requests through a local challenge-response mechanism by pressing a button or providing some other proof of physical presence (e.g., a fingerprint reader) to activate a service. According to the FIDO Alliance website, Google was one of the first companies to use authentication with tokens successfully.

Unlike traditional authentication methods, two-factor methods, such as those developed by Yubico [6], do without a central server. It was an urgent concern of Jakob Ehrensward, company CTO, to develop a decentralized authentication mechanism that does without shared secrets that communication partners need to safeguard. At the same time, however, he wanted it to be possible to use arbitrary services while ensuring the anonymity of the users (i.e., enabling any number of identities for any user).

In 2014, the FIDO Alliance simultaneously completed version 1.0 of the Universal Authentication Framework (UAF) and Universal 2nd Factor (U2F) protocols. In the years that followed, numerous client and server implementations appeared. At the same time, the Alliance gathered new members from the IT security, finance, software, and biometrics sectors. The first test and certification program appeared in 2015, followed by mobile implementations for iOS and Android, but also using contactless approaches with Bluetooth and near field communication. Microsoft started with Windows 10, and the Japanese provider NTT Docomo enabled its 65 million customers to log in without passwords.

FIDO2

In November 2015, Allianz submitted the design for FIDO2 to the World Wide Web Consortium (W3C). Just three months later, the W3C announced the establishment of a working group. Their goals were to promote standardization for strong authentication mechanisms of web browsers and websites based on the FIDO2 web APIs.

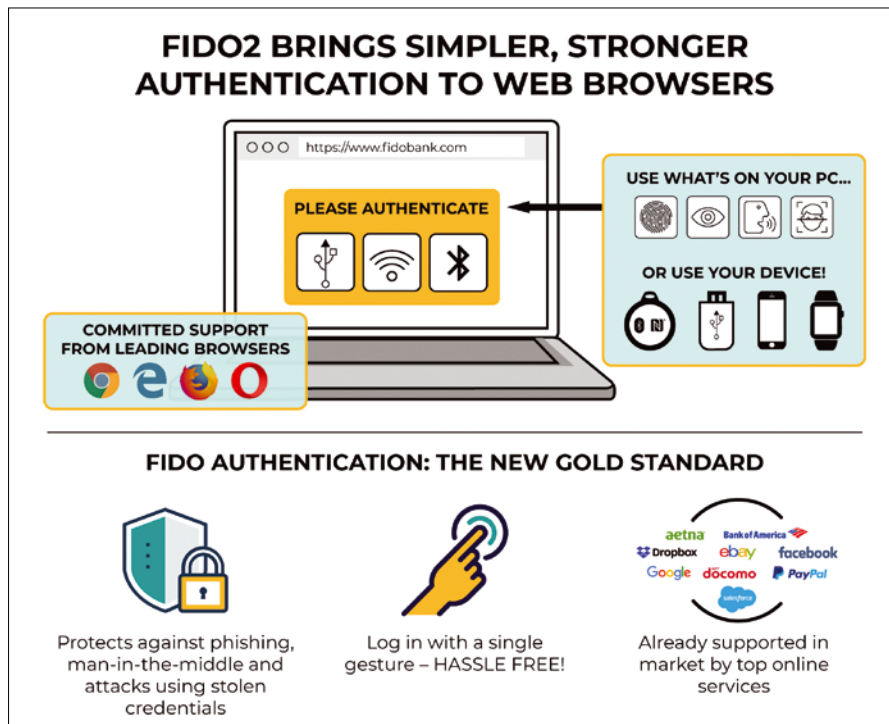


Figure 2: On the client device, the CTAP library uses an authentication method, whereas the browser uses WebAuthn to ensure secure communication with a web application or platform. Ideally, the user simply chooses a previously set up authentication method. Image © <https://fidoalliance.org>

The goal of the Web Authentication Working Group is to create a client-side API for web applications and to make e-payments with biometrics and other secondary factors both more secure and simpler, because most smartphones already have fingerprint readers and similar modules on board.

This scheme ensures greater user satisfaction and protects the authenticity of the credit card holder better than ever before, say FIDO and EMVCo, whose six members (American Express, Discover, JCB, Mastercard, UnionPay, and Visa) oversee "... the requirements for global interoperability between chip-based payment applications and acceptance terminals to enable secure contact and contactless transactions and other emerging payment technologies" [7].

In late 2016, the FIDO Alliance became the largest ecosystem for authentication standards. More than 200 certified solutions are now available, and thanks to Facebook integration, more than three billion people have access to secure login procedures. In March 2018, the W3C finally presented version 2 as a W3C Candidate Recommendation [8].

In addition to the FIDO classics UAF and U2F, FIDO2 now includes the Client to Authenticator Protocol (CTAP) and the WebAuthn protocol as specifications for integration in browsers and web applications [9] (Figure 2).

Operations

The entire operation needs to be simple and transparent for the user. Figure 3 shows the two variants, with a biometric feature on a smartphone (top) and USB tokens, smart cards, embedded secure elements, or trusted platform modules (bottom) as local device authentication. In the background, the new standard uses public key cryptography, which in combination with the local 2FA mechanisms also seems to arm the procedure well against any kind of phishing.

In the simplest case, the client device registers with the server using a public key. From that point on, the local device authentication can activate the private key to carry out the desired authentication.

Usually the server will prompt the user to select an appropriate authentication method and device for both sides. The client library then generates the special key pair for this combination

of local device, online service, and user account. The public key, also linked to the user account, then ends up on the server. The server does not receive any information about the local links on the customer side, so anyone breaking in would not be able to find out which authentication type the customer has chosen.

In detail, the registration process is as follows: The server sends a challenge to the user to log in as before (i.e., on the device and with the procedure that allows the keys to be read). If the client succeeds, it signs the challenge with its private key and sends the whole thing to the server. The server then only needs to check whether the public key matches the signature and then logs the user on. Alternatively, it triggers a requested process. The user can also store several keys on suitable media, which can then be used as required.

Firefox

In early 2018, the developer versions of Firefox introduced support for the authentication process, and the technology is firmly integrated since Firefox 60 [10]. Also, the developers of the

Chrome (Google) and Edge (Microsoft) browsers [11] plan to support WebAuthn in the future.

Listings 1 and 2 show JavaScript examples from a post by Mozilla developers J.C. Jones and Tim Taubert [12] that initializes a USB U2F token and requests public keys on the client. All you need is the local hardware (e.g., the USB token or a fingerprint reader).

The `navigator.credentials.create()` and `decodeCredential` functions of Listing 1 request a new FIDO credential by enabling all compatible devices and keeping them waiting for user input. The credential is created as soon as the user has authenticated successfully. The `decodeCredential` function creates a key pair and transmits the public key (in plain text). When generating the key, the URL call and a timestamp link this user, this website, and the authentication. Listing 2 shows the process for an existing credential when the user visits the website again.

For Developers

The FIDO Alliance developer resources web page [13] proves to be a

good starting point for developers, with direct links to the specification and explanations for the protocols, platforms, and functions involved (e.g., the `navigator.credentials.get()` and `navigator.credentials.create()` functions demonstrated in Listings 1 and 2).

On Android, `org.fidoalliance.intent.FIDO_OPERATION` handles all the functions (see chapter 6 of the W3C specification, or chapter 7 for iOS). The Android client API has defined this intent and it comes with a full set of functions for login, registration, and changes. Google's developer resources [14] prove to be just as helpful in the WebAuthn case as those of GitHub, where more and more projects are implementing the functions (e.g., in Python [15] and with YubiKeys).

Yubico tokens appear to be a good choice for a quick start. Both the YubiKey NEO and FIPS support FIDO and are National Institute of Standards and Technology (NIST)-certified for use in safety-critical environments.

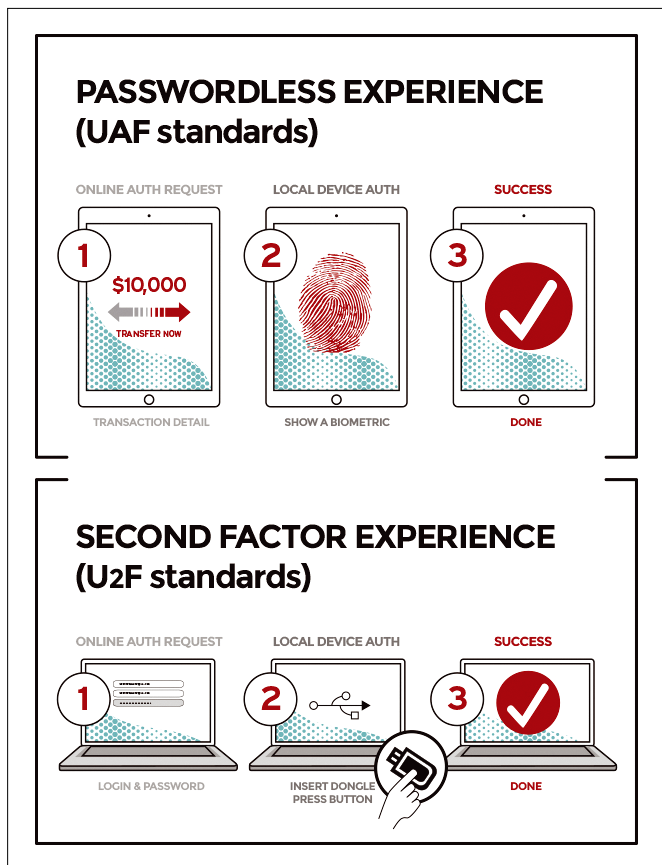


Figure 3: The FIDO alliance promises easy authentication in the future. Image © <https://fidoalliance.org>

Listing 1: Creating a Credential and Keys

```
01 /* (CC BY-SA 3.0:
    https://creativecommons.org/licenses/by-sa/3.0/) */
02
03 const cose_alg_ECDSA_w_SHA256 = -7;
04
05 /* The challenge must be produced by the server */
06 let challenge = new Uint8Array([21,31,105,<[...]>]); /*
29 more random bytes generated by the server */
07
08 let pubKeyCredParams = [{
09   type: "public-key",
10   alg: cose_alg_ECDSA_w_SHA256
11 }];
12
13 let rp = {
14   name: "Test Website"
15 };
16
17 let user = {
18   name: "Firefox User firefox@<example.com>",
19   displayName: "Firefox User",
20   id: new TextEncoder("utf-8").encode(
    "firefox@<example.com>")
21 };
22
23 let publicKey = {challenge, pubKeyCredParams, rp, user};
24 navigator.credentials.create({publicKey})
25   .then(decodeCredential);
```

Server Side

The FIDO Alliance points to the large number of servers that support FIDO, each of which offers its own configuration procedure. However, the Alliance

offers to confer FIDO Universal Server certification [16] on the servers.

Secure Sockets Layer (SSL) is a basic requirement, but without a database – or an LDAP user directory – nothing works right now. Yubico lists the steps necessary for LDAP integration under YubiAuth [17], and the LDAP wiki [18] indicates a starting point for CTAP. A demo project is available on GitHub for FIDO2 servers [19].

Hardware

A partner of the FIDO Alliance, StrongKey (formerly StrongAuth) [20], advertises its Enterprise Solution Key Appliance with high data security (Figure 4). For \$3,000, the FIDO Fast Start offer meets NIST requirements and is “designed specifically for government agencies.” If you ignore all the marketing and buzzword bingo, the StrongKey feature list looks good.

A Fast Start test seems to be the cheapest way in: Web apps developed for this purpose must support Chrome 50 or newer, must not use a captcha or one-time password, and must be limited to one URL – reloading external resources

Listing 2: Querying the Public Key

```
01 /* (CC BY-SA 3.0:
    https://creativecommons.org/licenses/by-sa/3.0/) */
02
03 /* The challenge must be produced by the server */
04 let challenge = new Uint8Array([42,42,33,<[...]>]);
    /* 29 more random bytes generated by the server */
05
06 let key = new Uint8Array(<[?]>); /* Retrieve Key Handle
07
08 let allowCredentials = [{
09   type: "public-key",
10   id: key,
11   transports: ["usb"]
12 }];
13
14 let publicKey = {challenge, allowCredentials};
15
16 navigator.credentials.get({publicKey})
17   .then(decodeAssertion);
```



Figure 4: The StrongKey appliance integrates FIDO2. Image © strongkey.com

A Webzine for High-Performance Computing Specialists



ADMIN
Network & Security

If you work with high-performance clusters, or if you're ready to expand your skill set with how-to articles, news, and technical reports on HPC technology.

admin-magazine.com/hpc

is prohibited. This setup should work fine with a Yubikey token or a smart-phone or laptop with biometric devices.

Criticism

FIDO2 seems to be a good combination of usability, security, and standardization. Supported by large corporations and organizations (e.g., from the credit card industry), the standard could spread rapidly – especially with the availability of ready-made, inexpensive hardware appliances.

The open source communities around Mozilla and Google (Chrome and Android) and the Apple and Microsoft communities also have done their homework and are driving the standard forward. For companies, especially, the combination of biometrics – despite all the inherent technological weaknesses – and tokens seems attractive, especially in the ongoing trend toward web services and cloud apps. However, the financial sector is likely to be the driving force. Presumably, many banking and credit card functions will be available soon with WebAuthn, as well.

Weaknesses

Because FIDO2 and WebAuthn authentication occurs in the browser, attacks on the browser are likely to remain the focus. According to the FIDO Alliance, WebAuthn is protected against man-in-the-middle and phishing attacks.

Expected problems are also hidden in the use of biometrics and credential recovery: The use of biometrics in place of passwords is risky. Iris scans, fingerprints, or keyboard and typing behavior should only replace the username, not the password. If one fingerprint is compromised, the user has only nine others. Moreover, users leave their fingerprints everywhere in everyday life, present their iris to every high-definition camera, and demonstrate their typing behavior to every web service.

Apart from these fundamental concerns, biometrics seem to be used sensibly in this case. The fingerprint only gives the authorization to generate the key within the presence display.

The problem of common password procedures (“Can I trust the server?”) can only be solved with individual passwords per service. At the end of the day, the WebAuthn procedure does a good

deal to help – without a password manager – but it makes credential memory a single point of failure that is likely to be the target of attacks.

Computer-independent analog storage of long, cryptic passwords generated per service is often even more secure, but who wants to put up with that today? WebAuthn’s approach is therefore going in the right direction: more security with appropriate usability. But it also has weaknesses.

Lost Token?

One inherent problem is the loss of a token, which raises the question of what recovery functions are offered by the online store. Even before WebAuthn, crypto guru Bruce Schneier warned [21] that any service that implements security questions (e.g., asking for the mother’s maiden name to reset the password) can also dispense with password rules.

With WebAuthn, only a new account – creating a new credential with a new

token – seems to be the solution. How the operators put this into practice will probably only be shown by implementations on bank and credit card sites, which is where you separate the chaff from the wheat today (e.g., with requirements for passwords or online banking procedures).

Compared with the long-known vulnerabilities of the popular mobile transaction authentication number (mTAN) method [22], however, FIDO2 is a great achievement with a huge amount of potential. The new standard is also unlikely to remove completely the need for web users to use passwords, TANs, and cryptographic keys. ■■■

Author

Markus Feilner has worked with Linux since 1994 as an author, trainer, consultant, and journalist. Today he heads the documentation team at Linux Distributor SUSE in Nuremberg, Germany.



Info

- [1] Fido the dog: [https://en.wikipedia.org/wiki/Fido_\(dog\)](https://en.wikipedia.org/wiki/Fido_(dog))
- [2] Abraham Lincoln’s dog: <https://americacomesalive.com/2013/07/17/abraham-lincolns-dog-fido/>
- [3] FIDO Alliance: <https://fidoalliance.org>
- [4] Fidonet: <https://www.fidonet.org>
- [5] FIDO Alliance members: <https://fidoalliance.org/participate/members-bringing-together-ecosystem/>
- [6] Yubico: <https://www.yubico.com>
- [7] EMVCo: <https://www.emvco.com/about/overview/>
- [8] W3C WebAuthn standards: <https://www.w3.org/TR/webauthn/>
- [9] Vision of the FIDO Alliance: <https://fidoalliance.org/approach-vision/>
- [10] Firefox and WebAuthn: <https://www.cnet.com/news/firefox-moves-browsers-into-the-post-password-future-with-webauthn/>
- [11] Chrome, Edge, and WebAuthn: <https://www.telecompaper.com/news/chrome-edge-firefox-to-support-new-log-in-standard-webauthn--1239550>
- [12] Instructions for Mozilla and WebAuthn: <https://hacks.mozilla.org/2018/01/using-hardware-token-based-2fa-with-the-webauthn-api/>
- [13] FIDO Alliance developer resources: <https://fidoalliance.org/participate/developers/>
- [14] Google’s developer resources for FIDO: <https://developers.google.com/android/reference/com/google/android/gms/fido/Fido>
- [15] Yubico Python script: <https://github.com/Yubico/python-fido2/blob/master/examples/credential.py>
- [16] Report from the RSA conference: <https://fidoalliance.org/rsa-conference-wrap-up-successful-launch-of-fido2-authentication-with-more-to-come/>
- [17] YubiAuth LDAP setup: https://developers.yubico.com/yubiath/LDAP_Setup.html
- [18] CTAP with LDAP: <http://ldapwiki.com/wiki/Client%20To%20Authenticator%20Protocol>
- [19] FIDO2 server: <https://github.com/apowers313/fido2-server-demo>
- [20] StrongKey appliance: <https://strongkey.com/key-appliance/>
- [21] “The Curse of the Secret Question” by Bruce Schneier: https://www.schneier.com/blog/archives/2005/02/the_curse_of_th.html
- [22] mTAN: https://en.wikipedia.org/wiki/Transaction_authentication_number

LINUX UPDATE

Need more Linux?

Our free Linux Update newsletter delivers insightful articles and tech tips to your mailbox twice a month. You'll discover:

- 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



www.linux-magazine.com/newsletter

MakerSpace

GoPiGo3 robot vehicle
with a Rasp Pi core

Go Pi Go!

The GoPiGo3 kit provides components and software for a small robot car with a Raspberry Pi brain. *By Martin Mohr*

The easiest way to learn programming is hands on. The GoPiGo3 robot kit [1], based on a Raspberry Pi, supports programming with the Bloxter graphical programming language, up to and including the development of complex applications with Python, Node.js, or C/C++.

The GoPiGo3 design makes a solid, robust impression (Figure 1). The basic frame comprises two Plexiglas plates

screwed together with threaded bolts. The direct-drive motors are mounted on stable metal brackets. With a little skill, assembling the kit takes less than an hour. The manufacturer provides detailed instructions on their homepage [2] to make it easy to assemble the robots.

The power supply for the robot uses rechargeable batteries; two sets are necessary if you don't want to wait several hours for the batteries to recharge between programming sessions.

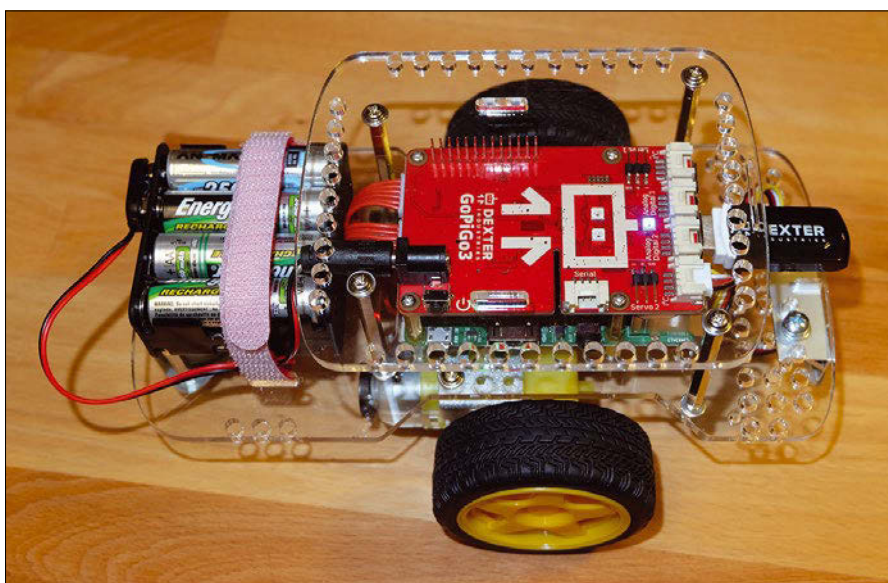


Figure 1: The fully built GoPiGo3 offers everything a robot car needs. With optional accessories, you can add sensors and actuators to the structure as desired.



Figure 2: You program the robot in a web front end. The GoPiGo3 website takes you to the various programming environments and documentation.

To conserve battery power while programming the GoPiGo3, you can connect a USB power supply to the Rasp Pi.

A detailed manual [3] helps you set up the robot. After booting, the model creates an unencrypted WiFi network with the *GoPiGo* SSID, which you log into on

your PC. Please note that you might lose access to the Internet as a result. Now enter the address <http://bloxter.com> in a browser. The web server running on the kit responds to this URL (Figure 2).

A simple menu under the *Drive* button lets you control the car remotely

for the first function tests. The *Learn* button provides a variety of exercises to help you familiarize yourself with graphical programming in Bloxter [4], a simple graphical environment based on Scratch.

Under *Code in Bloxter*, you develop your own programs. With a little experience you will see which Bloxter functions are available in highlighted menu items. The individual commands are structured like pieces of a puzzle that you can arrange into a program by dragging and dropping into place. Only pieces that work together can be clicked together, so it's impossible to make a mistake in syntax. The examples in this article were created in Bloxter.

Alternatively, the editor under *Code in Python* lets you control the robot with a lower-level programming language. As with Bloxter, you run the programs by clicking the hardware

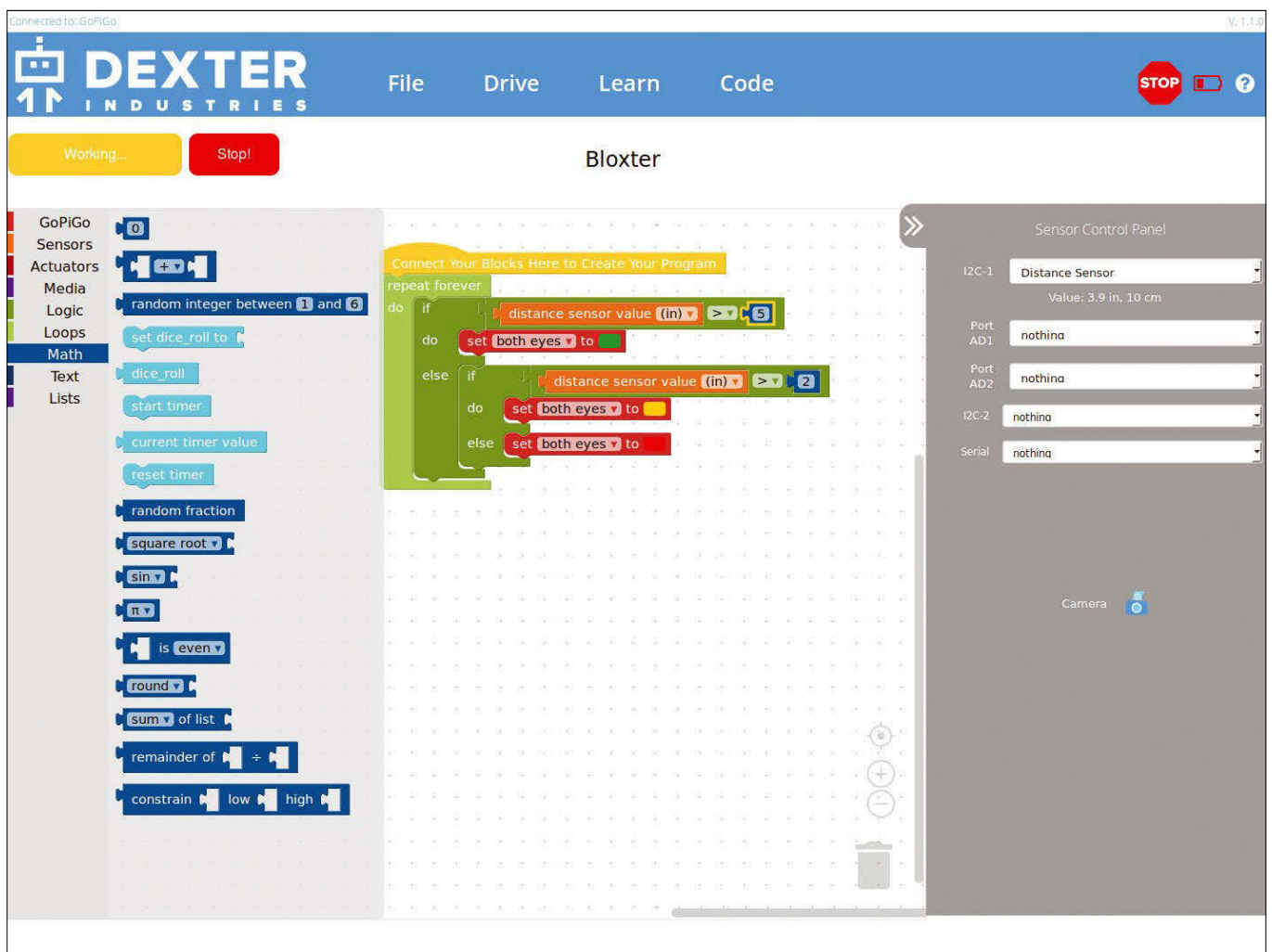


Figure 3: The first sample program in Bloxter: The GoPiGo3 changes its eye color according to readings from the proximity sensor.

Start button. The GoPiGo_Python_Examples folder contains many examples that can be used as a basis for your own experiments. I use two of these programs to illustrate the programming in detail.

Proximity Sensor

In this example, the GoPiGo3 uses its proximity sensor to control “eye” color as a function of measured distance. The two LEDs on the top side of the board act as eyes. If the sensor detects a large distance in front of the robot, the eyes glow green. If the sensor detects an obstacle, the eyes become yellow. If space is cramped, red eyes signal danger.

Figure 3 shows the required Bloxter program. The first thing you notice is that you don’t have to import any additional libraries. All the required functions are available by default. The individual function groups in the left sidebar differ in color: Light green stands for loops, orange for reading sensors, and red for actuators. The outer loop ensures that the robot executes the

program infinitely. If necessary, you can interrupt an endless loop by clicking on *Stop*.

The first if block in the loop checks whether the distance to the next obstacle is greater than 5 inches and switches the eyes to green if the result is positive. (If desired, the value output by the sensor can be converted to centimeters.) The constant number for the comparison can be found at the top of the *Math* functions list. If the sensor outputs a shorter distance, the next if block decides whether the eyes should glow red or yellow. That is all it takes to convert the values measured by the proximity sensor to colors.

Avoiding Obstacles

To really bring the GoPiGo to life, you have to engage the motors to move the robot. The spectrum of possible algorithms for this programming task is enormous, but many roads lead to Rome. In the next example, I explain the basic functions of setting the GoPiGo3 into motion while avoiding complicated routines.

The program (Figure 4) starts with a *while* loop that queries the proximity sensor. The robot moves forward until it encounters an obstacle (i.e., until the proximity sensor returns a value less than 5 cm). For your first driving attempts with the GoPiGo3, you should always set the speed to *slow*. The robot is quite robust, but you should still make sure it doesn’t go around a corner too fast or fall down stairs.

As soon as GoPiGo detects an obstacle, the program leaves the *while* loop and starts a routine to dodge the obstacle: Like an evasive maneuver test (i.e., moose test), the robot turns 90 degrees to the left and moves 15 cm forward, turns 90 degrees to the right and moves 15 cm forward, and so on; the rest of the commands are self-explanatory. As soon as the robot has driven completely around the obstacle, it moves a little farther and then stops.

The robot is best driven on a smooth surface – but not too smooth – otherwise the tires will lose traction. Carpet is not a recommended substrate. Depending on the structure of the carpet,

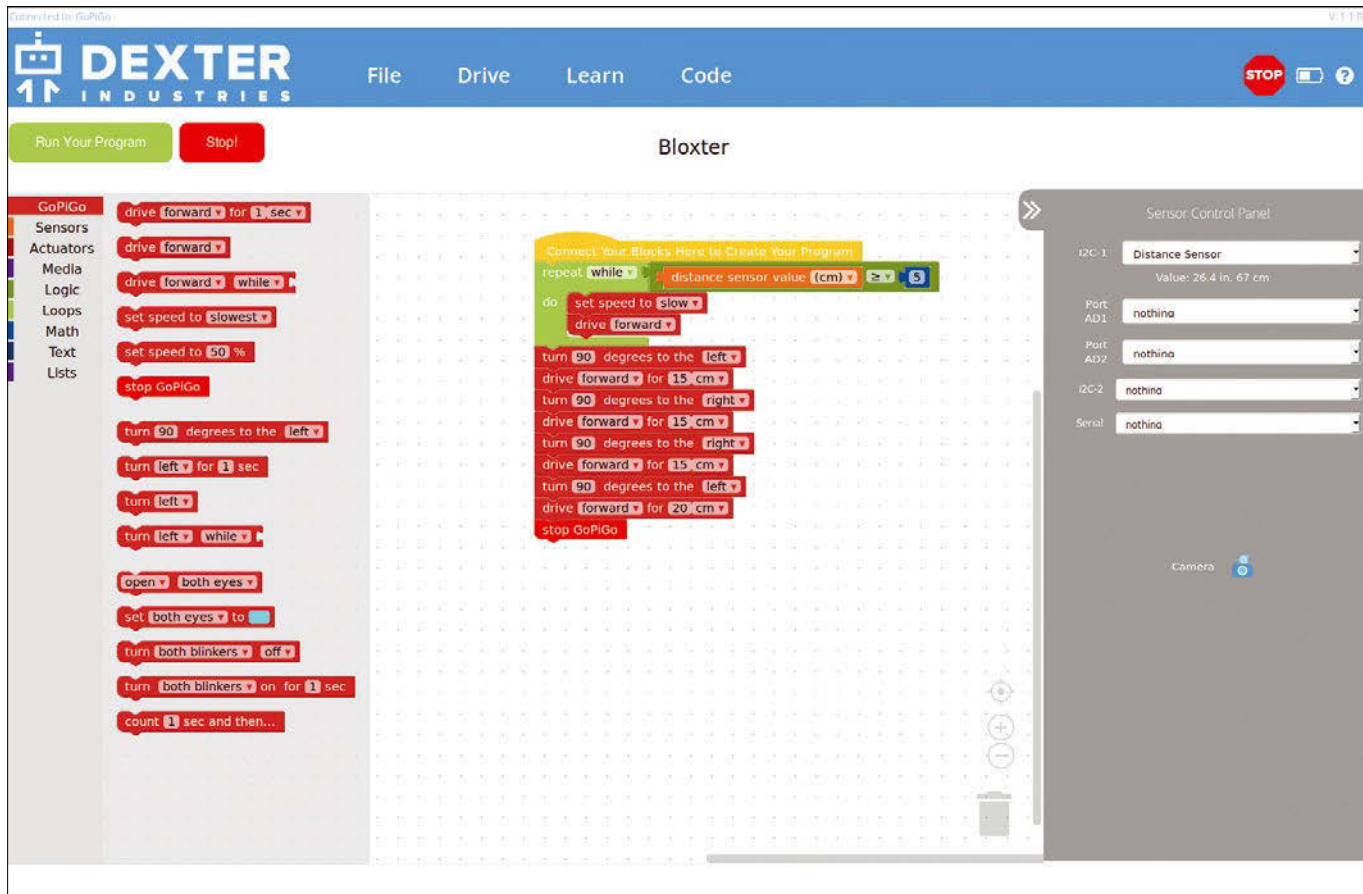


Figure 4: The Bloxter program instructs the GoPiGo3 to drive straight ahead until it encounters an obstacle. It then tries to avoid the obstacle and continue its journey.

Table 1: Accessories

| Component | Description |
|--|--|
| Light & Color Sensor | Detects brightness and different colors |
| Line Follower | Lets the GoPiGo3 follow a line (black adhesive tape) on the floor |
| Grove Infrared Receiver | A receiver for IR remote control |
| Infrared Remote | IR remote control |
| Loudness Sensor | Lets the robot react to noises |
| Grove Buzzer | A simple buzzer |
| Grove Button | A button |
| Speaker | Speaker for the Raspberry Pi |
| Grove LED (red, white, blue, green) | Light-emitting diodes |
| Servo Package | A servomotor and mounting hardware |
| IMU Sensor | Detects motion, orientation, and position for building a balancing robot |
| Temperature, Humidity, & Pressure Sensor | Gathers environmental data |

the support wheel can sink so far that it restricts the robot's mobility. You will notice that the robot no longer turns precisely (e.g., 60 degrees instead of 90 degrees) or that it moves forward at an uneven speed.

Hardware, Modules, and Software

The GoPiGo3 is available in different versions. The simplest is the GoPiGo3 Base Kit for \$99 (CAN\$132; UK£111, EUR107, excl. VAT) [5]. It includes the controller board, the chassis, the motors with encoders, the battery holder (without batteries), and a package with screws – more than you need for assembly. Even if you lose a screw, you should still be able to complete the robot.

The next stage, the GoPiGo3 Starter Kit for beginners (\$200; CAN\$ 263; UK£ 228, EUR239, excl. VAT) costs about twice as much [6]. The package contains all the components of the GoPiGo3 Base Kit, plus a Rasp Pi 3, additional servos, a proximity sensor, an SD card with DexterOS preinstalled, an 8GB USB stick, and a USB power supply for the Raspberry Pi.

The GoPiGo3 Starter Kit gives you all the components you need to start programming directly in Bloxter or Python. The free Raspbian for Robots download [7] lets you connect your robot to the Internet and program in C, Java, or Node.js. In addition to the different versions, you can purchase more sensors and accessories in the shop [8]. If you are not in the US, Dexter Industries has a number of distributors around the world [9].

Table 1 shows an overview of additional parts available individually, so you can assemble the appropriate modules for your own experiments. The various shops also offer various kits specially tailored for classroom situations.

Conclusions

The GoPiGo3 helps students and hobbyists familiarize themselves with robotics. The Bloxter development environment makes it easy to convert your ideas into code without extensive programming knowledge. Because of the robust design, the model is well equipped for demanding everyday use in schools.

The GoPiGo is primarily intended for students as a learning platform; therefore, the manufacturer offers detailed documentation on the web. The

GoPiGo Tutorials & Documentation page [10] has various how-tos, from setting up the robot to programming in Python or Node.js. Additionally, a number of complete projects are described that guide your build, step by step. The texts are in English and always well illustrated, so getting around should not be too difficult.

All told, the GoPiGo3 is an impressive set of hardware for would-be robot enthusiasts. ■■■

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.

Info

- [1] GoPiGo3: <https://www.dexterindustries.com/gopigo3>
- [2] Building instructions: <https://www.dexterindustries.com/GoPiGo/get-started-with-the-gopigo3-raspberry-pi-robot/1-assemble-gopigo3/>
- [3] Getting started: <https://www.dexterindustries.com/GoPiGo/get-started-with-the-gopigo3-raspberry-pi-robot/2-connect-to-the-gopigo-3/>
- [4] Bloxter: <https://www.dexterindustries.com/introducing-bloxter-the-graphical-programming-language-for-learning-coding-with-robots/>
- [5] Base Kit: <https://www.dexterindustries.com/shop/gopigo3-robot-base-kit/>
- [6] Starter Kit: <https://www.dexterindustries.com/shop/gopigo-beginner-starter-kit/>
- [7] Raspbian for Robots: <https://www.dexterindustries.com/howto/install-raspbian-for-robots-image-on-an-sd-card/>
- [8] Shop: <https://www.dexterindustries.com/shop/?orderby=price>
- [9] Distributors: <https://www.dexterindustries.com/distributors/>
- [10] Tutorials and documentation: <https://www.dexterindustries.com/gopigo3-tutorials-documentation/>

MakerSpace

An input device for keyboard shortcuts

Simplifying Shortcuts



For even simpler keyboard shortcuts, the prospective Palitra allows you to create simple macros with a programmable USB device. *By Bruce Byfield*

The keyboard is by far the most efficient way to interact with a computer. Pressing keys is faster than a mouse or trackball and reduces repetitive stress injuries, as well. For many, these advantages make learning keyboard shortcuts well worth the effort. Unfortunately, most non-gaming keyboards lack separate programmable keys, and shortcuts are often complex key combinations. Michele Balistreri's forthcoming Palitra offers simpler shortcuts: a programmable USB device for storing several dozen shortcuts. Palitra is scheduled for a crowdfunding campaign to assist manufacturing [1].

When he was 13, Balistreri first encountered open source software in a Corel Linux CD included in an Italian magazine. Since then, he has contributed to MPlayer and the icculus fork of Quake 2, briefly maintained KDE packages for Gentoo Linux, developed embedded software, and designed hardware.

Working as a fashion photographer inspired Balistreri to develop Palitra. He explains, "As a photographer, I spend a lot of time retouching, using a Wacom tablet and a keyboard for shortcuts. However, the typical keyboard is rather large and some combinations are hard to activate with one hand (especially being

left handed). Often, my hand gets in the way of the stylus [for the tablet], and I cannot keep it constantly on the keyboard. To avoid shoulder ache and time wasting, I decided that I needed a small input device, which could be placed next to the tablet and programmed with all the shortcuts I need."

Balistreri searched for an existing device like Palitra, "but they looked more flashy than useful – and most were over \$200. So, I decided to develop Palitra" (Figure 1).

Balistreri sees distinct advantages to releasing his work as open hardware. Open hardware, he says, offers "access to a lot of great engineers and developers who are willing to answer your questions and give you feedback. When you ask questions on public forums, you can show your design or code. This means you can get very direct answers instead of abstract advice or pointers. [As a result], you can more readily build a community of passionate people." At any rate, he adds, if you are crowdfunding, "the results should be a public good."

Admittedly, releasing open hardware makes knockoffs more likely. However, Balistreri discounts that problem. "If a project is successful, it will be copied regardless of its openness," he observes. "By that time, however, you have already

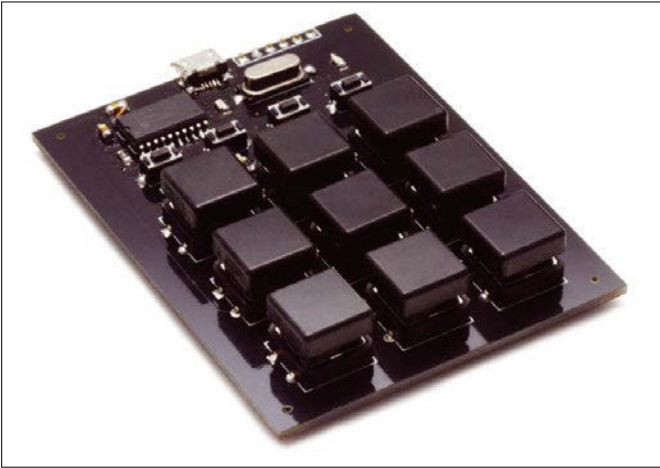


Figure 1: Palitra is a portable keyboard designed for shortcuts.

cuts I use the most in photography, like zooming and changing brush size. Then I thought I could add more shortcuts and added the concept of pages to organize shortcuts and added a single button that cycled between them and four LEDs to show the active page.” The result, after some soldering, was the

oversized prototype (Figure 2). “Luckily,” Balistreri continues, “I had the very fitting microcontroller, the PIC18F14K50, so I quickly developed the first firmware. In its very first revision, the shortcuts were hardcoded; then I developed a way to configure them, and my wife developed a desktop application to do that in a user-friendly way” (Figure 3).

With these basics ready, Balistreri began using his creation. However, in the early prototypes, changing pages required cycling through them, so he added a button to activate each one. Then it was time to take the project public. Creating a repository [2], he remembers, “I formalized the schematics, designed a custom PCB board, and replaced the small switches with larger ones with a nice plastic cap. In addition, I replaced the soldered USB cable with a micro-USB connector, making the device more portable and less prone to failure.”

More recently, Balistreri has added a bootloader, making the firmware upgradable via USB, and his wife, Kseni Balistreri has updated the software with the ability to save profiles. “This is a great feature,” Balistreri notes, “because with it you can quickly reconfigure the entire device, making it easy to create profiles for each application you use. It will be possible to share profiles, and there will be some default ones in the desktop app.”

As I write, Balistreri is working on the last step. Although his father made a

Product Development

Thinking about the project, Balistreri originally figured out that he needed a “small keypad where I could assign short-

created a vibrant community around your project and have recouped your R&D costs. All you need to do is keeping being better than the knockoffs, and you’ll continue to be successful.”

Shop the Shop  shop.linuxnewmedia.com

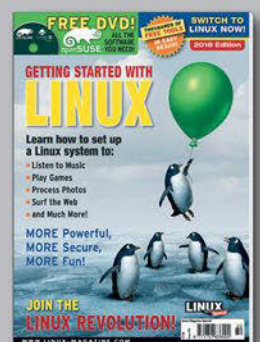
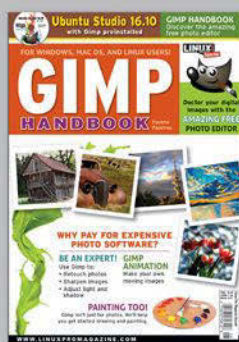
Discover the past and invest in a new year of IT solutions at Linux New Media’s online store.

Want to subscribe?

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

 shop.linuxnewmedia.com

DIGITAL & PRINT SUBSCRIPTIONS



SPECIAL EDITIONS



Figure 2: The first prototype was designed for function, not portability.

case from wood and aluminium for prototypes, the result – while attractive – is too expensive to produce commercially (Figure 4). As a result, he is currently finalizing the design of an injection-molded enclosure and developing a 3D-printed enclosure in the interim.

The result is a device that a computer recognizes as another input device. Designed for Linux, Mac OS, and Windows, Palitra should sell for \$30 or less, if all goes according to plan. Balistreri plans to emphasize its use for photography, but mentions that it should also be useful for “graphic design, vector art, and

even software debugging” – particularly on minimalist keyboards where function keys require pressing several keys. It would also be ideal for hardcore gamers, who could load separate profiles for each of their favorite games. With the help of simple scripts, Palitra might even be useful for inputting text, allowing the addition of boilerplate information like a biography or copyright notice with the press of a single button.

Coming Next

Balistreri has been working with a few prototypes. He now faces the challenge



Figure 4: The prototype enclosure shows Palitra to advantage, but is too expensive for the final product.

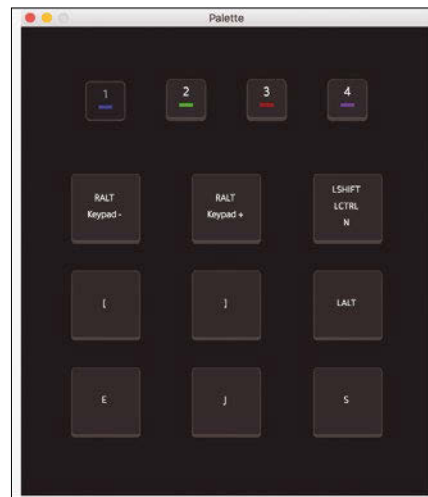


Figure 3: The prototype software for customizing Palitra’s firmware.

of manufacturing Palitra as inexpensively as possible – a goal that can be challenging because of the relatively low initial production quantities. “I want to keep Palitra very affordable,” he says, “because a lot of people need it.”

Sooner or later, too, Balistreri expects a request for a wireless version. However, a wireless device will require a different microcontroller and PCB design, so for now, it is not his main focus. Other features are likely to follow as users make requests, but he is determined to avoid any enhancements that interfere with the main purpose or make the device less portable. His hope is that other photographers will find Palitra as indispensable as a spare battery for their cameras.

Palitra is not the only solution for creating simple macros. Graphic tablets routinely have programmable keys, as do high-end gaming keyboards. Similarly, the Keyboardio Model 01 has several methods for setting up dozens of keyboard shortcuts. However, those who have discovered the advantages of shortcuts, can never have too many – and the shorter, the better. With its simplicity of elegance, Palitra seems likely to be welcomed as another example of how open hardware is changing the way we use computers. ■■■

Info

[1] Palitra crowdfunding campaign:
<https://www.crowdsupply.com/bitgamma/palitra>

[2] Palitra on GitHub:
<https://github.com/bitgamma/palitra>

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

CONTEGIX
AGILITY BUNDLE
AN INTERFACE TO MANAGE DEPLOY
POINT CLICK DEPLOY
LEARN MORE

ADMIN
Network & Security

ADMIN Update – Hottest Links

- Monitoring and Alerting with TICK-Stack
- Bleedingbit: Two New Bluetooth Vulnerabilities
- Intel Chips Smashed by PortSmash
- Just-in-time Administration in Active Directory
- Consul

Highlights

Monitoring and Alerting with TICK-Stack
If you are looking for a monitoring, alerting, and trending solution for large landscapes, you will find all the components you need in the TICK Stack. (more)

Intel Chips Smashed by PortSmash
New exploits take advantage of side-channel Simultaneous multithreading capabilities. (more)

Bleedingbit: Two New Bluetooth Vulnerabilities
New vulnerabilities for Bluetooth Low Energy chips made by Texas Instruments (more)

Just-in-time Administration in Active Directory
Just-in-time administration affords admins the ways and means of enforcing the validity period for extended privileges. (more)

Consul
Modern-day DevOps techniques bring automation to everyday tasks, such as service discovery, monitoring, and

Further Reading

- Performance Tuning with the top
- Oracle Goes Cloud Native
- Chinese Spy Chip in US Servers?
- Storage Virtualization and High Availability
- Is North Korea Hacking US ATM Machines

SC18
NOVEMBER 11-16, DALLAS
The International Conference for High Performance Computing, Networking, Storage, and Analysis
LEARN MORE →

BeoWorld at SC18

Going to SC18 in Dallas and want to meet the folks who started it all? Come to the annual Beowulf Bash, the party thrown BY the HPC community FOR the HPC community. Monday night, November 12, 9pm to midnight at Eddie Deen's (944 S Lamar, 5 minute walk from convention center).
[Click here to learn more!](#)

www.admin-magazine.com/newsletter

MakerSpace

Simple wearable projects
with your Android phone

Tech on Tap



Create some interesting wearable projects with an Android device, an armband case, headphones, and MIT's free App Inventor package. *By Pete Metcalfe*

A wearable solution with voice recognition and speech-to-text capabilities can be used by anyone who needs information while their hands are busy or when they can't take their eyes away from their current task. Some examples include recipes for chefs, instruction manuals for mechanics, or directions for a cyclist.

In this article, I create an Android application that recognizes a spoken keyword that is searched in a text file; text-to-speech software echoes the result of the search. To create the Android app, I use MIT's App Inventor package, a free web-based Android app development tool you can use to create applications in a graphical environment.

For this project, you need a smartphone armband case and a set of headphones with a built-in microphone (Figure 1), as well as a Google user account [1] and a desktop or laptop computer to use App Inventor.

App Inventor

App Inventor [2] is a web-based app creation tool for Android devices with a graphical programming environment. App Inventor has two main screen modes: In Designer mode, you lay out the Android app in the Viewer screen, and in Blocks mode, you build the logic. On the right side of the top menu-bar, the *Designer* and *Blocks* buttons allow you to toggle between these two modes.



Figure 1: Hardware needed for this project: an Android smartphone mounted in an armband and headphones with a mic.

In Designer mode, you create an app layout by dragging a component from the Palette window onto the Viewer window. For the visuals on this application, you will use the following components from the User Interface section of the Palette: *Button*, which initiates speech recognition, *Label*, which first shows the *Hit the Talk Button* message and then shows the result of the speech recognition, and *ListView*, which shows the lines of the data file.

Also, you need some non-visual components. In the Media section of the Palette, drag the *SpeechRecognizer* and *TextToSpeech* components into the Viewer window and add the *File* component from the Storage group (Figure 2).

The Components window lets you rename or delete components. When you select a component, the Properties window shows the component's editable features. In this example, I renamed the button *BT_Speak* and changed the *BackgroundColor*, *FontSize*, *Width*, and *Text* fields from their defaults (Figure 3).

Logic

Once the layout design is complete, you can add logic by clicking on the *Blocks* button in the top menubar. Logic is built by selecting an object in the Blocks window and then clicking on the specific block you want to use (Figure 4).

App Inventor is pretty amazing when it comes to fast prototyping. This entire app only requires one variable and four when blocks (Figure 5). The first step is to load the text file by calling the *when Screen1.Initialize* block. The *when File1.GotText* block loads the text file data into the global variable (THELIST) and popu-

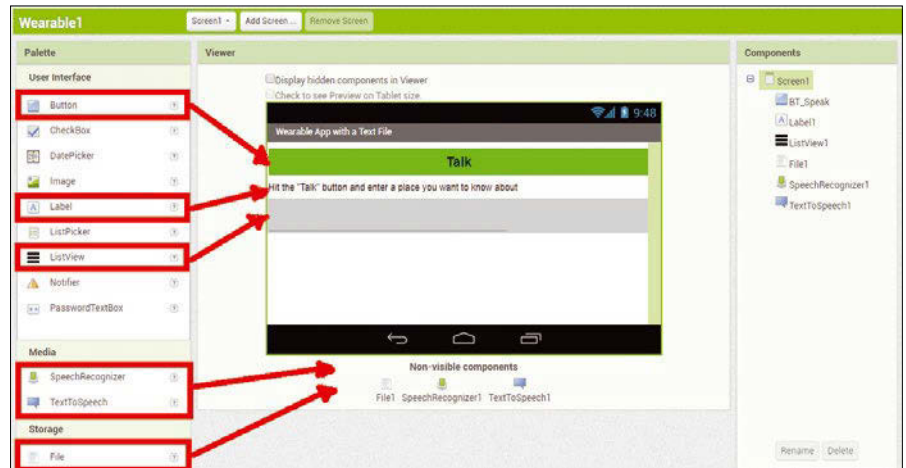


Figure 2: App layout.

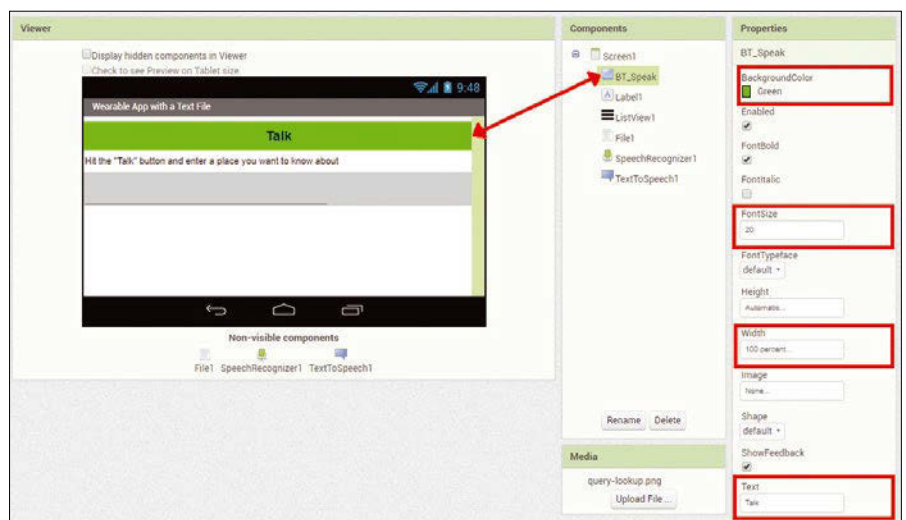


Figure 3: App components and properties.

lates the *ListView* component.

The *when BT_Speak.Click* block is activated on a button push and starts the speech



Figure 4: Logic blocks added to the Viewer window after clicking *Screen1*, the current project.

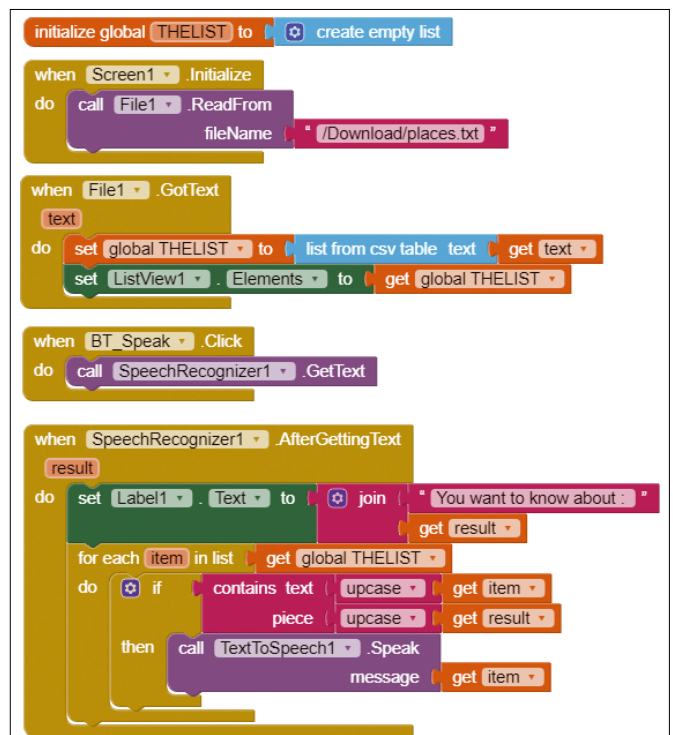


Figure 5: Logic blocks filled out for the current project.

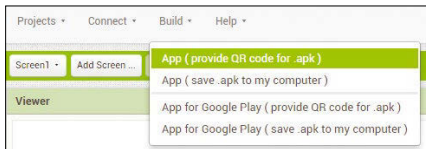


Figure 6: Build options.

recognition block. The final block, *when SpeechRecognizer1.AfterGetting-Text*, shows the result of the speech recognition in a label and it checks whether it is in the global variable. If the result is found, a text-to-speech message is generated with the full line of text.

Data File

The data file I used, *places.txt*, is a list of some local landmarks, but you could use a number of other data choices, such as friends' addresses, recipe ingredients, or favorite restaurants. For readability, I recommend you put the keywords at the beginning of each line; for example:

```
"Hope Bay has a sandy beach with..."
"The Glen is a horseshoe-shaped valley..."
"Isaac Lake is a bird sanctuary with..."
"Oliphant is great for kite surfing..."
...
```

For this example, the file was saved in the Android's *Download* directory; this location needs to match the `fileName` entered in the `File1.ReadFrom` block definition (i.e., `/Download/places.txt`) in the first `when` block.

Compiling and Running the App

After the screen layout and logic are complete, the *Build* menu item compiles the app, which can be made available as an APK downloadable file or as a QR code link (Figure 6).

Once the app is installed on the phone, pushing the *Talk* button opens the Google speech recognition dialog. If you've spoken a valid keyword, you should hear the line from the data file (Figure 7). You can update the data file without making any changes to the app.

Final Thoughts

In this example, I used a simple text file, but you could enhance the code to support multifield CSV files, cloud services, HTTP requests, or Google Maps. ■■■

Info

- [1] MIT App Inventor sign-in: <http://ai2.appinventor.mit.edu/>
- [2] App Inventor info: <http://appinventor.mit.edu>

Author

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

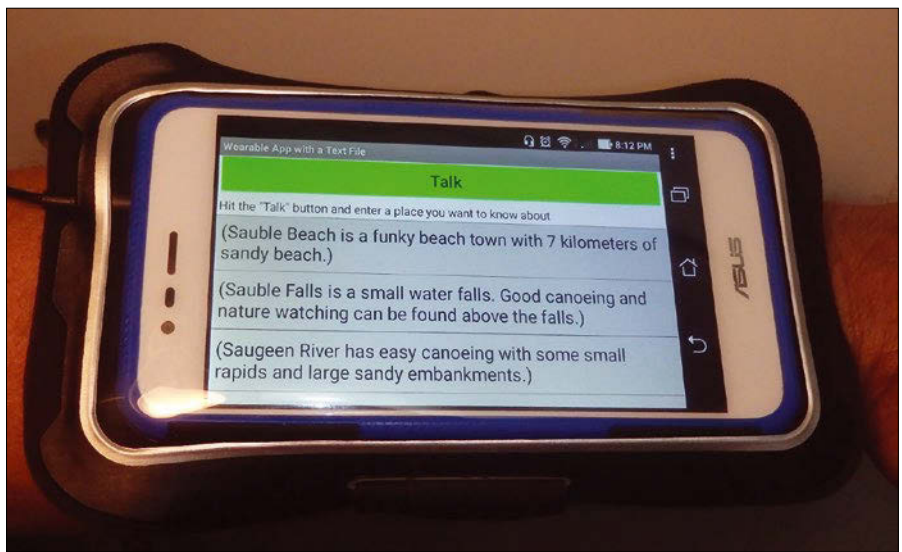


Figure 7: Final Android app listing the elements in the text file. In this example, the query results are only returned as an audible. You could very easily add another Label and show the query results here.

Celebrating 25 Years of Linux!

ORDER NOW
Get 7 years of *Ubuntu User*
ON ONE DVD!



When OS vendors got the idea of offering bounties to security specialists who discover vulnerabilities, they thought they had hit upon a perfect formula for staying ahead of hackers, spies, and cyber thieves. But all formulas have an expiration date. The game has changed now; plain old vendors and their white-hat security experts have a little competition.

According to many reports, the NSA, and similar agencies around the world, have a big appetite for vulnerabilities that lead to exploits, and they are bidding up the bounties. Are they putting the public at risk by keeping these vulnerabilities secret? In this month's Linux Voice, Markus Feilner explores the new world of the military-malware complex.

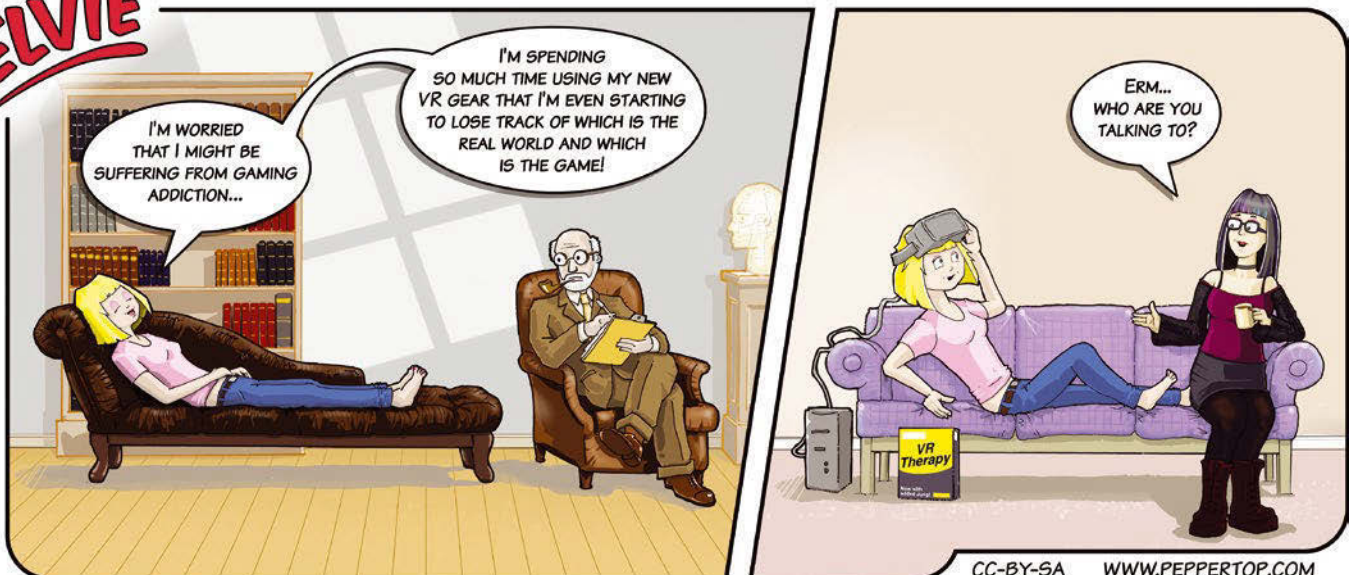
Also in this month's issue, we introduce you to Retroshare, a security communication platform that operates without a central server using peer-to-peer networking techniques. We also help you get started with Plasma – KDE's powerful but little-understood desktop.



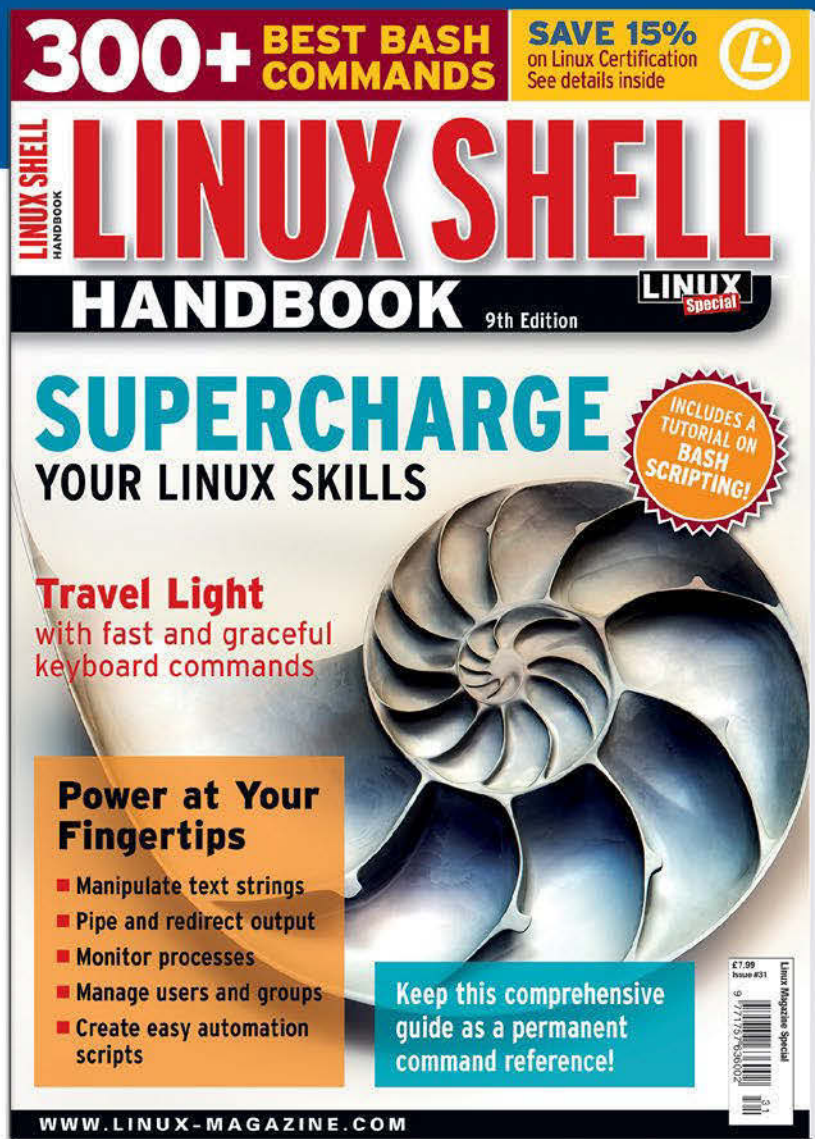
LINUXVOICE ▶

| | |
|--|-----------|
| Doghouse – IoT | 71 |
| <i>Jon "maddog" Hall</i> | |
| In conjunction with Caninos Loucos, maddog helps develop a hardware platform for the Brazilian IoT project. | |
| Retroshare | 72 |
| <i>Erik Bärwaldt</i> | |
| Communication on the Internet is not as secure as you might think. We show you how to eliminate curious eavesdroppers. | |
| Military-Malware Complex | 76 |
| <i>Markus Feilner</i> | |
| Modern cyberwarfare and its resulting monetary allocations have significantly impacted the exploit market. | |
| FOSSPicks | 84 |
| <i>Graham Morrison</i> | |
| This month Graham looks at draw.io, musikcube, Whalebird, and more. | |
| Tutorial – Plasma | 92 |
| <i>Paul Brown</i> | |
| Get started with KDE's powerful desktop. | |

ELVIE



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

MADDOG'S DOGHOUSE



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

In conjunction with Caninos Loucos, maddog helps develop a hardware platform for the Brazilian IoT project. BY JON "MADDOG" HALL

A hardware platform for the Brazilian IoT program

I have been spending a lot of time in Latin America recently, in particular Brazil. The object of my visits has been the Brazilian Internet of Things (IoT) program and the project Caninos Loucos ("Crazy Canines," in Portuguese).

The Brazilian government formulated a plan a couple of years ago to stimulate IoT in Brazil. They feel that the IoT industry could generate more than \$200 billion in goods and services, and perhaps as much as 20 percent of the Brazilian Gross Domestic Product (GDP) will depend on technologies used in IoT.

Some time ago, I cofounded Caninos Loucos [1] along with Dr. Marcelo Zuffo of LSITEC, an NGO associated with the University of Sao Paulo.

At first, the project was just to allow high school and university students to purchase a single-board computer similar to the Raspberry Pi, which (for many reasons) can cost more than \$120 in Brazil.

Then, the objective was to create a computer that could also easily be used for commercial products: one that could operate at commercial temperatures, have good electrostatic discharge (ESD) protection, possess a good number of standard buses, and to do this in an "open" way.

The project was also to stimulate other digital design projects and products, to give interesting work to computer engineers and electrical engineers, and to create more manufacturing jobs inside of Brazil and Latin America.

Then Dr. Zuffo became aware of the Brazilian IoT program. Traveling to the country's capital, he convinced that program that Caninos Loucos could design, develop, and manufacture a line of computers that would be the hardware platform.

We determined that we needed three systems to start, and since the project was "Crazy Canines," we named them after dogs.

The Labrador (the friendly dog) was the first. It was targeted to be a design platform, with a System on a Module (SoM) "core board" that contained the ARM four-core 32-bit CPU, a GPU, memory, and flash. The SoM talked to a motherboard that contained all the I/O controllers, power management, ESD, and connectors through a 204-pin DIMM connector. More interested in having the system be flexible and useful than just "inexpensive," we aimed for a balanced system in CPU, memory, and I/O. We also believe that, by using the SoM technology, we can safeguard the design of motherboards designed and manufactured today by standardizing on the 204-pin bus structure. Core

boards of the future (with faster and more capable system units) will be able to drive the motherboards of today.

The second system, the Pulga (the "flea"), was a tiny sensor computer, using an ARM M4F processor, 128K SRAM, and 512K flash on a SoM board, with the communications (Bluetooth 5.0 Mesh or LoRaWan) and sensors being on another board with a power source (battery or uber-capacitor) sandwiched between the two boards. The whole system would be the size of a 10-cent Brazilian coin (about the size of a dime in the US) and should be able to run for at least six months on a watch battery.

The third board was a blockchain router based on the Subutai Blockchain Router designed by my company OptDyn. OptDyn contributed the design to the project (the design is also available on GitHub) and cooperated with LSITEC in the further development of the board. The blockchain router is a powerful broadband router, NAS server (RAID 0-10 configurable), 802.11b/g/n, and IoT gateway. As an IoT gateway, you can mount an Arduino shield or use the 40 GPIO pins (compatible with the Raspberry Pi) to control IoT projects in your home or business. The SoC used in this system has an ARM four-core, 64-bit processor, as well as two two-core real-time processors and a field-programmable gateway array (FPGA). The system has up to 8GB of RAM for the operating system and 8GB of RAM for the FPGA.

Caninos Loucos has plans to develop 64-bit Labradors and more designs for the Pulgas, but we want to bring these designs to market to get the IoT program (software and applications) started.

It is not the intent of Caninos Loucos to turn LSITEC into a manufacturing plant. As these designs are created and perfected, LSITEC will be contracting with private companies to produce these systems under license and to make them available to anyone who wants them at the lowest possible prices. We also intend on embracing other motherboards designed and manufactured by other people and companies (and in other countries) to extend the capabilities of these boards.

The reception we have had for the design and production of these boards has been astounding. We hope to have these in large-scale manufacture in time to have them underneath your Christmas tree. ■■■

Info

[1] Caninos Loucos: <http://caninosloucos.org>

Secure Communication

Communication on the Internet is continuously subject to the risk of being intercepted. We show you how to eliminate curious eavesdroppers.

BY ERIK BÄRWALDT

The Retroshare communication platform licensed under the GPL ensures a secure exchange of data at all levels. The program is easy to use and offers some special features, such as integration of the Tor network. The software is also available for download for all popular operating systems.

Communication on the Internet always entails the risk of someone listening in. Many users thus look for easy-to-use encryption methods. However, because of the numerous forms of communication on the web, each with its own protocol and data transmission path, no standard solution suffices.

Moreover, most services are based on client-server architectures, wherein the server, as the central data repository, is the preferred target of attack. Errors in server configuration – over which the user usually has no influence because it is often a centralized service operated by a commercial provider – can be misused as a gateway for malware and espionage software.

However, peer-to-peer networking is far more secure. To avoid the massive installation overhead caused by the different protocols and transmission paths, users are now turning to the graphical Retroshare platform, which combines multi-functional communication in a uniform interface.

Operations

Retroshare [1], under development since 2006 and licensed under the GNU GPL, is a friend-to-friend network (F2F) [2]. In principle, it works like a peer-to-peer network without a central server instance, the difference being that the participants communicate with each other on a basis of trust.

To ensure a certain basic trust, communication between two Retroshare participants requires the exchange of their RSA keys generated by the software itself. The application also supports what is known as turtle hopping [3].

This form of communication distributes files and information across multiple computer systems on the Retroshare network, even if the

source and target computers are not directly known to each other or have not authenticated each other with a mutually stored public key.

However, a maximum of seven forwarding nodes between the source and target computers knows the systems from which they receive data and information directly and the neighboring nodes to which they pass content. This process ensures a chain of paired computers that know and have authenticated each other so that man-in-the-middle attacks can be ruled out. Because the transport route and the data are encrypted by OpenSSL and OpenPGP, data security is very high.

Versatile

Retroshare is available in many places; for example, you will find it in the software repositories of many Linux distributions: ALT Linux, Slackware, Gentoo, and ROSA Linux maintain their own packages, and the project's website [1] provides various links to repositories for Arch, Debian, CentOS, Fedora, Mageia, Ubuntu, openSUSE, and others.

If you fail to find a suitable package despite this selection, you can pick up two Applimages and the source code, which can be launched on practically any Linux system; they share the ability to run only on 64-bit hardware. The binaries maintained in the package managers, on the other hand, are also usually suitable for 32-bit environments. Two Applimages exist for ARMv7- and ARMv8-based computers like the Raspberry Pi, as well.

Installation

To install distribution-specific packages, follow the instructions in the Download section on the project's website. In both cases, the software adds a launcher to the start menu. To load the Applimage initially, it needs execute permissions, which you can assign with the `chmod +x` command.

After starting, Retroshare displays a graphical wizard in which you enter some personal data to create a user account, including a username, password, and a name for your node in the initial screen. Finally, you generate a PGP key with a pre-

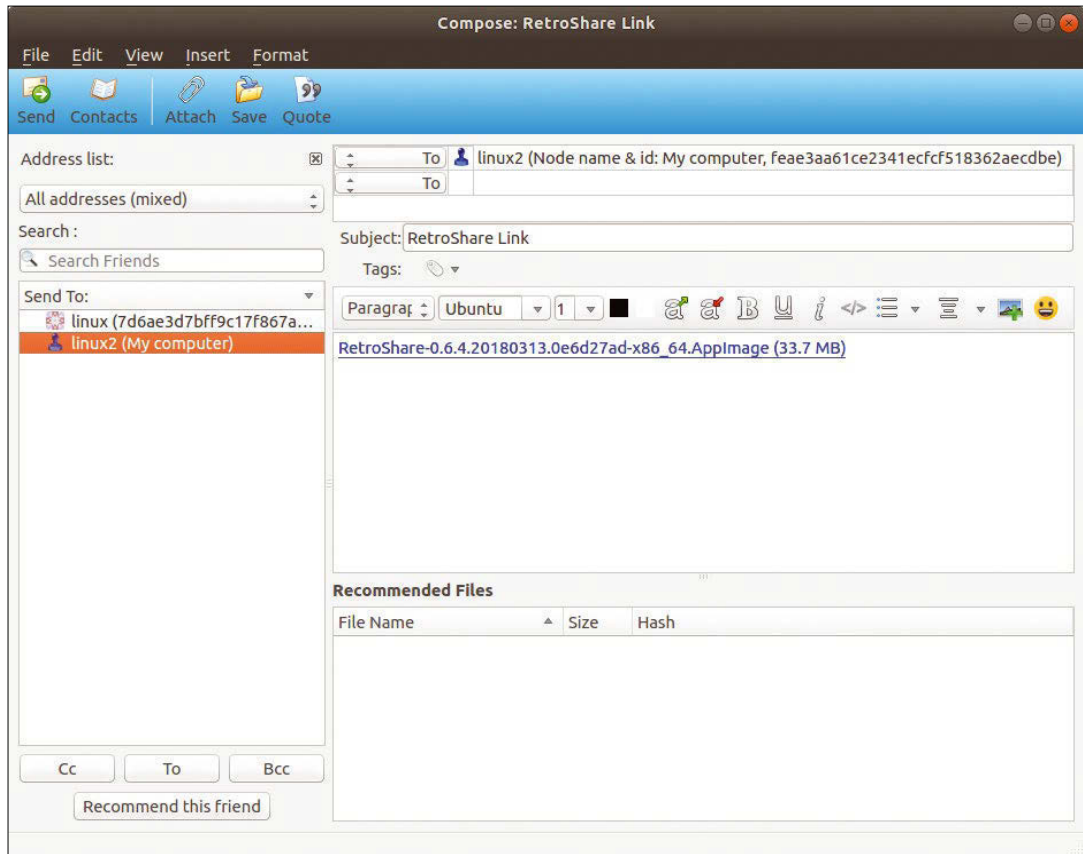


Figure 3: Sending links to shared files with an automatically generated email.

A chess game that allows two Retroshare users to play over the web is available as an extension, as well. Please note that some plugins are still experimental, so you might not want to use them in production environments for stability reasons.

Figure 4: Tracking the download of shared files to the target computer in real time.

Data Exchange

Retroshare supports file transfer, although it is not directly comparable to a normal cloud system. Whereas cloud applications focus on the

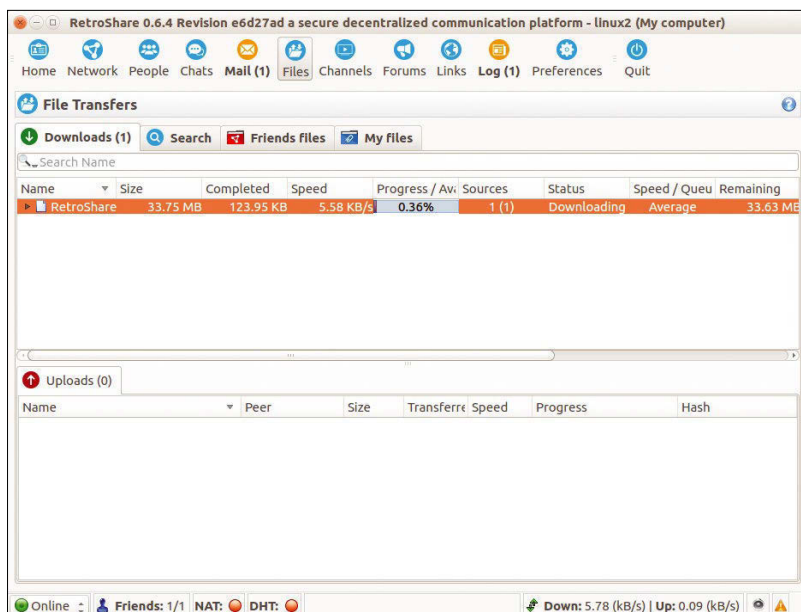
synchronization of data across several computers on an intranet, the Retroshare file sharing platform is set up for exchanging data. Therefore, you can send files not only to your Retroshare partners, but also to nodes that are not in direct contact with your computer system. In all cases, the software guarantees complete encryption of the data – even during transmission.

To exchange data, first click *Files*. In the following window, switch to the *My Files* tab and double-click the *My Files* entry in the list view. A submenu pops up and you can right-click it to select *Open Folder* from the context menu.

Your file manager then opens the Retroshare share directory, to which you then drag the files to be shared. If you double-click on the share folder in the Files dialog, the files stored in it appear in a list view. If this is not the case, click the *check files* button.

Select the files you want to share and right-click them to open the context menu. Select the *Send retroshare Links* option, which opens the mail window with links to the files in the message area (Figure 3).

On the left side of the mail window, you select the recipients to receive the message and start the download by clicking on one of the file links. A separate download window opens in which the recipients can confirm that they want to copy the file by clicking on *Download*.



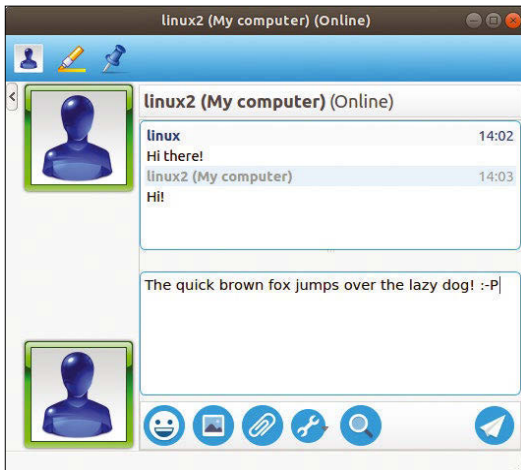


Figure 5: The integrated chat feature lets you transfer files to communication partners.

When the recipients open the Files dialog and click on the *Downloads* tab, the current downloads and the completed downloads are shown in a list. Note that downloading large files can take some time because of integrity checks (Figure 4).

The recipient then finds the files in the *My Files* tab of the Files dialog. They can be viewed there by right clicking and selecting *Open File*. Retroshare uses external system programs to manage different kinds of files.

In the *Downloads* tab of the File Transfers window, right-click one of the files and select *Open Folder* to open the folder containing the shared files. You can also create additional sharing folders (e.g., for specific groups). The software lets you exchange files directly between two directly connected nodes and by way of encrypted, anonymous tunnels that cover up to seven hops.

Note that for hops across multiple instances, all nodes involved must be switched on and logged on to the network: If a node is deactivated on the transmission path, the path to the target computer is interrupted and the transmission fails.

The chat function is only suitable for transmitting individual files directly to communication partners. In the chat window, which you open by double-clicking on a friend in the Network window, a small buttonbar at the bottom of the chat history offers a paper clip icon for sending attachments, which you can select conveniently in a file manager. If you write a message offline, the

Info

- [1] Retroshare: <http://retroshare.net/>
- [2] F2F network: <https://en.wikipedia.org/wiki/Friend-to-friend>
- [3] Turtle hopping: https://en.wikipedia.org/wiki/Turtle_F2F
- [4] Tor network: <https://www.torproject.org/>

software sends the message or attachment as soon as a connection is established. Circulars to several recipients can also be provided in this way (Figure 5).

Tor Network

For those of a particularly cautious nature, for whom Retroshare’s encryption mechanisms are not sufficient, the developers also offer a version that exclusively uses the Tor network as a transport layer [4]. You can find this version on the project website as an Applmage for 64-bit hardware.

The Tor version of Retroshare is usually missing from the software repositories of the popular Linux distributions. Unlike the traditional version, it integrates a preconfigured Tor client, so you don’t have to complete any additional installation work. The user interface of this version is the same as that of the conventional version.

The Tor variant is particularly suitable for those who maintain contacts with others they do not know personally. Retroshare automatically establishes the connection to the Tor network. In the program window, you will find visual confirmation by the Tor onion symbol at the bottom (Figure 6).

Conclusions

Retroshare supports secure communication with end-to-end encryption, password protection, and, if required, Tor network integration. The software has a stable, intuitive interface, which makes the program ideally suited for any kind of communication between trustworthy partners who value maximum security. ■■■

Figure 6: For anonymous connections with participants not personally known to you, Retroshare supports communication via the Tor network.



Pretty Complex

Modern cyberwarfare and its resulting monetary allocations have significantly impacted the exploit market, but where does that lead?

BY MARKUS FEILNER

In October 2018, at a European Union data privacy conference, Apple CEO Tim Cook attacked competitors, Facebook and Google. According to Cook, Facebook and Google's business models had become a "surveillance industry" and likened their services (unlike Apple's devices) to a "data industrial complex" [1].

Analysts quickly agreed that Cook's underlying intention might have been to spin better public relations for Apple after several unfriendly articles about Apple's tax-evasion strategies. However, Cook's comments provoked more than the expected rebukes from Google and Facebook – it drew the attention of people who have been following American politics and military strategies since WWII.

Many were reminded of US President Dwight D. Eisenhower's legendary farewell address in 1961. The former five-star general and commander of Allied forces in Europe warned the American pub-

lic about the risks and dangers the "military-industrial complex" (installed during and after WWII) posed for the free and democratic Western world. Even though Eisenhower wrote his speech in a time when tanks and oil dominated warfare, some of his words remain as powerful today as they were during the cold war [2].

Today, this military-industrial complex has advanced into a new domain: If data is the new oil [3], then access to data is crucial for corporate and national prosperity. Detailed information about people and companies is considered the decisive factor in elections, polls, and nearly every product's selling point. The more the government and companies know, the better, as witnessed by the size of the NSA's hard drives and data centers. Needless to say, the Big Four (Facebook, Google, Apple, and Amazon) hold many keys in this game, which also makes them a target for hackers – much like certain operating systems are targeted.

What Is a Zero Day?

A "zero day" is a software bug, a vulnerability that has not been patched by the vendor. It usually is unknown to those responsible for closing it, which would prevent third parties from gaining unauthorized access. It's like knowing about a house's unlocked back door. At the moment, nothing serious has happened; whoever finds the unlocked back door might just inform the owner, who could then lock the door or fix the lock. Zero day references the fact that the home owner or software vendor has zero days of knowledge about the flaw. Once informed, the time counter starts, and the zero day is not a zero day anymore.

On the other hand, a less honest person might sell that back-door knowledge, maybe anonymously, to a third party, perhaps on the darknet. Let's assume the back door was locked, but a lock-picking burglar found a way to open the lock with a specially created key (i.e., through "advanced" technology). The third

party might be ready to pay even more now, so the burglar might sell the key and the information to them. In IT, this key would be called an exploit, in this case a zero-day exploit. Using this exploit to enter the back door would make this a zero-day attack. With a Windows, Apple, or Android operating system monoculture, imagine huge neighborhoods of homes that all share the same bad, exploitable back-door lock: Once uncovered, every house is vulnerable until every single owner has fixed the flaw.

Of course, this kind of knowledge might be valuable to more than just criminals searching for treasure. A widespread open backdoor gives governmental intelligence an untraceable means for slipping through security to install spyware, bugs, or other surveillance equipment – and there is a lot of proof that these intelligence services are busy now acquiring zero day exploits.

Data breaches happen: In September 2018, Facebook reported a cyberattack that affected 30 million users; just a few weeks later, Google admitted that its social network Google Plus had been compromised since March 2018. In both cases, “software glitches” had been exposed and used by hackers to access customers’ data.

Windows, Android, and iOS are known for their flaws and backdoors, as shown by a recently published German government report on IT security [4]. Hackers continue to reveal zero-day exploits (see the “What Is a Zero-Day?” box), like when SandboxEscaper recently disclosed a new and scary Windows problem [5]. Security holes like the new Bleedingbit vulnerability for Cisco, Meraki, Aruba, and other wireless access points [6] show that even hardware isn’t immune; this is a general IT problem, one that cannot be solved through open source alone.

The Exploit Market

Over the past decade, fewer and fewer exploits seem to be available, and at much higher prices than before. Depending on who you ask, you will hear different reasons for this. Activists and politicians claim that a financially well-equipped (read tax money) malware-digital complex keeps stockpiling and buying exploits so that there are simply no other influential buyers available. If a hacker finds a flaw, he’ll sell it to the military or an affiliated institution and happily receive a decent reward. Early in 2018, Motherboard published a 2015 letter [7] from the Israeli Ministry of Defense asking hackers for zero-day exploits (see Figure 1).

In the broader security world, few people were surprised by this letter being sent out to various companies. More interesting was its openness and clarity, since in 2015 it was not public knowledge that Western nations were involved in buying zero-day exploits. That knowledge came later, mostly in 2016 and 2017. In the US, the Vulnerabilities Equities Process (VEP) [9] was developed in 2008/2009 but was only unveiled in 2016, after the Electronic Frontier Foundation (EFF) filed a Freedom of Information Act (FOIA) request [10]. This was followed by a wave of ethical, philosophical, and political discussions circling around the question of whether a democratic state should engage in this type of activity.

Shadow Brokers and the VEP

In spring 2017, The Shadow Brokers (TSB), a hacker group said to have close ties to the Russian government, published several controversial NSA documents and tools. According to TSB, the NSA had been stockpiling exploits for Microsoft Windows and the international banking software SWIFT. TSB published proof of the claims, showing how US tax money had been used to put Win-

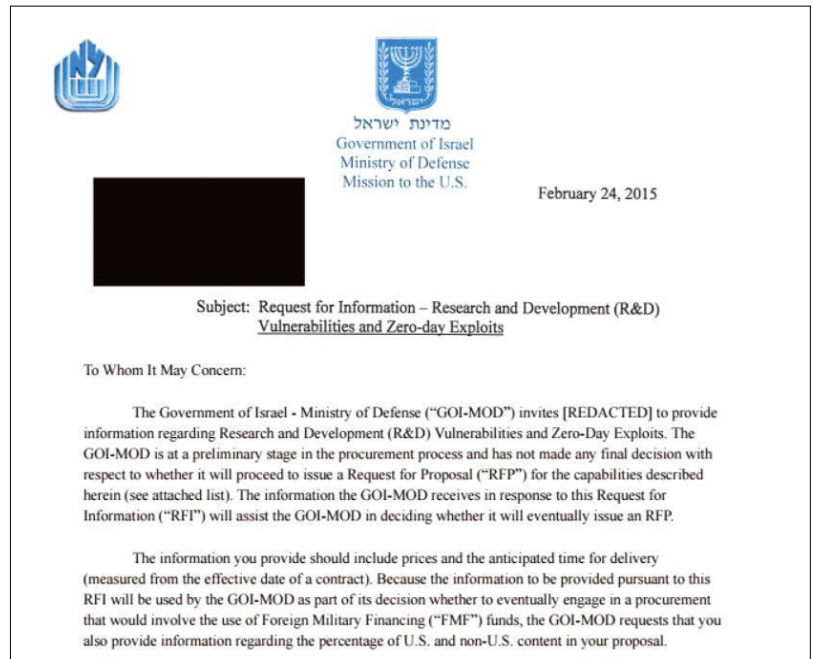


Figure 1: In 2015, Israel’s Ministry of Defense asked hackers to send them their newest zero-day exploits [8].

dows users and banking customers in danger of being compromised rather than protecting them by informing Microsoft and the banks. The NSA’s embarrassment over being hacked, which proved them incapable of protecting their own secret hacking tools, didn’t last as long, though.

As one of the consequences, the US government took a more transparent approach in explaining the VEP in November 2017 [11]. Today, an Equities Review Board (ERB) decides on actions. The ERB meets monthly (or in emergency situations), and its members come from the US departments of Treasury, State, Justice, Energy, Defense and Commerce, the Office of Management and Budget, the CIA, and the Department of Homeland Security. With the NSA as the executive entity, the ERB follows four steps:

1. Submission and notification
2. Equity and discussion
3. Determination to disseminate or restrict
4. Handling and follow-on actions

Criticism of the VEP noted a number of deficiencies, from non-disclosure agreements to insufficient risk ratings to special treatment of the NSA to a missing default disclosure policy. While on paper the standard action defaults to disclosure, there are too many options to circumvent full disclosure. Another unanswered question is whether these cyber weapons fall under the jurisdiction of any existing international arms treaties.

The Cold War Onward

If you take a deeper look at the history of cyberwarfare in the US, some surprising facts pop up. Perhaps the oldest record of an offensive cyber-

war attack by the US government dates back to 1982 – if you believe Thomas C. Reed, an Air Force secretary in the Reagan administration, who claimed that a Trojan in CIA-doctored software was responsible for blowing up a Siberian gas pipeline [12]. It’s a long read from there to Natanz, Iran, where alleged American-Israel cooperation

sabotaged a nuclear facilities’ centrifuges via Stuxnet [13], a malicious computer worm. Whereas Soviets denied US involvement in the 1982 explosion, the Obama administration half-heartedly admitted involvement in Stuxnet in Natanz (see the “Stuxnet and Wannacry” box for more information on two of the most successful

Stuxnet and Wannacry

Stuxnet targeted supervisory control and data acquisition (SCADA) and programmable logic controllers (PLCs). It was so sophisticated that even experts failed to believe their eyes when it was discovered in 2010. Stuxnet attacked SCADA systems in Iran, combining four zero-day exploits (e.g., in the Windows operating system) and targeting PLC in order to falsify sensor data. While SCADA provides GUI and high-level management, PLCs are usually the machine interfaces in industrial environments. With its roots in the 1960s, SCADA is basically used everywhere today, from power plants to any kind of industrial or commercial device or machinery. SCADA’s age is also its main problem: There are huge security issues, most of them systematic, in data encryption, verification, or a complete lack of any security layer.

Stuxnet consisted of three components: a worm, a link file, and a rootkit. It was transmitted even to air-gapped systems through compromised USB sticks. The 2016 movie, *Zero Days*, explains why it’s highly likely that Stuxnet was an Israeli attack with a US-built tool. The movie also includes the interesting diplomatic background story [17].

Analysts likened Stuxnet’s use to “opening Pandora’s box,” since it legitimized for the first time digital warfare through sophisticated malware that was constructed by intelligence services with visible results. Furthermore, the broad public analysis brought about massive publications on the attackers’ functions and proceedings. In late October 2018, news from Iran claimed a second similar attack, dubbing it Stuxnet II [18].

A few months after TSB released a bunch of the NSA’s secret backdoors and tools, a worldwide cyberattack, using the ransomware Wannacry, rendered hospitals (much of the UK National Health Service), institutions, and millions of computers unusable. Until May 2017, Wannacry is said to have successfully attacked several hundreds of thousands of PCs in hundreds of countries, causing up to billions of dollars of damage. The NSA had known about EternalBlue, a Samba flaw used by Wannacry, and deliberately not told Microsoft. Microsoft, however, discovered the backdoor on their own and fixed it for their most recent systems. However, not all customers had applied all necessary updates, and users of older versions were on their own for even longer. The USA, UK, and Australia said North Korea was the creator of Wannacry.

DoD Cyber Strategy

The Department of Defense (DoD) Cyber Strategy 2018 outlines the following core points [19]:

“First, we must ensure the U.S. military’s ability to fight and win wars in any domain, including cyberspace.

“Second, the Department seeks to preempt, defeat, or deter malicious cyber activity targeting U.S. critical infrastructure that could cause a significant cyber incident regardless of whether that incident would impact DoD’s warfighting readiness or capability.

“Third, the Department will work with U.S. allies and partners to strengthen cyber capacity, expand combined cyberspace operations, and increase bi-directional information sharing in order to advance our mutual interests.”

In addition, the strategy puts forth the following objectives:

- “1. Ensuring the Joint Force can achieve its missions in a contested cyberspace environment;*
- “2. Strengthening the Joint Force by conducting cyberspace operations that enhance U.S. military advantages;*
- “3. Defending U.S. critical infrastructure from malicious cyber activity that alone, or as part of a campaign, could cause a significant cyber incident;*
- “4. Securing DoD information and systems against malicious cyber activity, including DoD information on non-DoD-owned networks; and*
- “5. Expanding DoD cyber cooperation with inter-agency, industry, and international partners.”*

malware attacks with government ties). Wikipedia offers a long and interesting entry on US cyberwarfare history [14], including a timeline. Then there is Donald Trump's National Cyber Strategy from September 2018 [15] (see the "DoD Cyber Strategy" box). Last, but not least, the United States Cyber Command (USCYBERCOM) "has the mission to direct, synchronize, and coordinate cyberspace planning and operations to defend and advance national interests in collaboration with domestic and international partners" [16].

Within the vast body of literature regarding US cyberwar strategy, the recent German publication *Cyberwar – Danger from the Network*, by Constanze Kurz and Frank Rieger (from the Chaos Computer Club Germany and the renowned Netzpolitik blog) provides insight into modern cyberwarfare tactics and strategies [20]. Kurz and Rieger deal with a variety of topics, including Stuxnet and the impossible task of deterring enemies, and they explain in detail why classical warfare and intelligence work won't succeed in cyberspace, but will waste taxpayers' money to a previously unseen extent.

Often cyberattacks are an attempt to hide prior failed covert activities to prevent discovery by enemy intelligence services or – at worst – secret tools being compromised. Standard tools, like the Territorial Dispute function used in government malware, check for the presence of other malware on the system before taking further actions. Territorial Dispute is one of the first functions called by the malware, right after a successful system break-in. Like an intrusion detection system or antivirus software, Territorial Dispute scans the machine and its files and returns errors such as "go get help immediately", "friendly service", or "get out of here" – it's a scanner for other malware! Kurz and Rieger's story gets even crazier: Future high-end malware will wait for Territorial Dispute's scan and force it to return the "get out of here" value immediately as a defensive mechanism. For the attacking intelligence service, a failed attack is far less dangerous than being caught or having their tools compromised. Kurz and Rieger's book also gives insight into where all the money is going.

Money, Money, Money

There is a huge market of software developers, admins, hackers, and surveillance technology orbiting US military as well it's affiliated industries (Wikipedia lists BAE Systems, EADS, Leonardo, General Dynamics, Raytheon, and Thales, just to name a few). In the Europe, Lench IT's FinFisher [21], dubbed "Remote Monitoring and Deployment Solutions," has been the basis for many state-run Trojans deployed on Apple and MS Office, mostly by third-world regimes, but

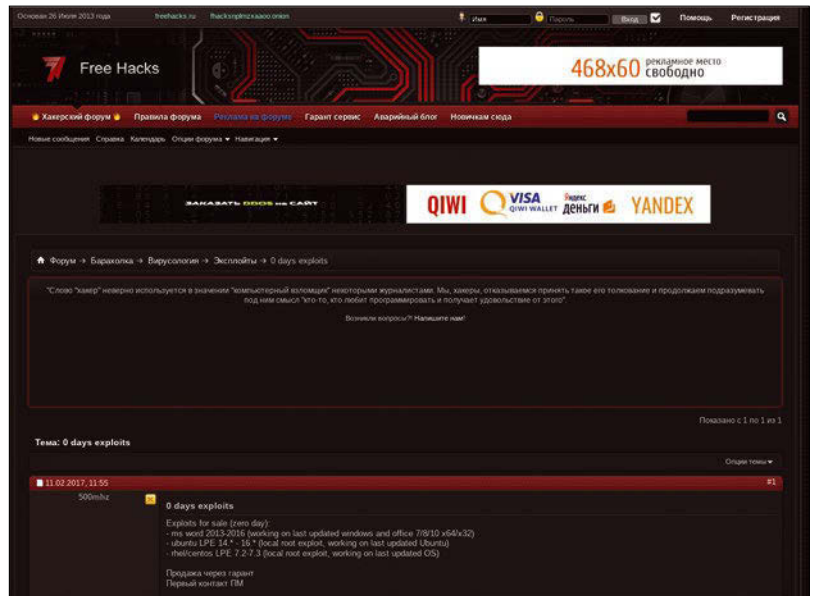


Figure 2: FreeHacks is one of the more commonly known exploit markets on the darknet. Knowing Russian might be helpful...

also by the German government. The tool seems to be quite efficient: In 2013, Reporters without Borders called it "Corporate Enemy of the Internet," and several human rights initiatives have criticized it and its customers harshly. The public controversy around FinFisher has resulted in a long and broad public discussion, leaving lots of traces of how the digital-military complex works and how governments fund "security" companies. The "enhanced remote deployment methods" that companies like FinFisher love to advertise are usually nothing more than exploits, usually zero-day exploits, which is where it becomes expensive. Most exploits are traded on the darknet on websites like the Russian hacking forum, FreeHacks (Figure 2), but these portals change quickly.

When you're shopping on the darknet, you better bring money. At the BOS Data Festival (BDF) 2015, Adriel Desautels explained [22]: "Zero days are used for very specific things. Things you usually don't get access to. The value is determined by operational need and window of time. The value of an exploit in the zero-day market is determined by target distribution. [...] The prices start at \$110,000 for a single zero-day, non-exclusive exploit and can range up to millions. [...] If you sell something to a customer for 50K in one year, and he comes back the year after, needs something similar, and pays 200K, you know he might be pretty powerful and has a valuable target." In the same presentation, Desautels gave his audience some peace of mind: "You're not the target: Service providers are a much easier way for the government to get your data. They will hand over data; the government doesn't need to pay to get ordinary people's data."

zerodium

Home For Researchers For Clients Careers About Us

We are Zerodium

The leading exploit acquisition platform for premium zero-days and advanced cybersecurity research.

"We pay BIG bounties, not bug bounties"

Program Overview

Latest Announcements

ZERODIUM is always improving its bug bounty program and payouts, and constantly expanding the list of eligible software. Our latest announcements and bounties can be found below:

Sep. 19, 2018 - We are acquiring pre-authentication RCE exploits affecting the following Routers: ASUS, Cisco, D-Link, Linksys, MikroTik, Netgear, TP-Link, and Ubiquiti. Exploits leading to authentication bypass or credentials disclosure are also accepted. Exploits relying on XSS or CSRF are not eligible.

[Read More](#)

Sep. 13, 2018 - ZERODIUM increases the payouts for various products including: Chrome, WordPress, Apache, and many others, and adds new entries to the program: nginx, Exim, WinRAR, 7-Zip, WinZip, cPanel, Webmin, Plesk, NetBSD, OpenBSD, and FreeBSD.

[Read More](#)

Aug. 23, 2018 - We are currently paying up to \$100,000 for code execution exploits affecting major file archivers: WinRAR, 7-Zip, WinZip (on Windows 10/8.1) or tar (on Linux). The exploit must rely on a common file archive format/extension and must have a reasonable file size. Valid until Oct. 31st, 2018.

[Read More](#)

Figure 3: Zerodium offers bug bounties.

The amounts Desautels mentions still seem credible, as high as they are. Zerodium [23], a major zero-day buyer, offers bug bounties on its website. One such bounty (Figure 3), dated August 23, 2018, offers "up to \$100,000 for code execution exploits affecting major file archivers: WinRAR, 7-Zip, WinZip (on Windows 10/8.1), or tar (on Linux)."

Exploit Lifetimes

Obviously, there's a lot of money in the game, and its origin seems pretty clear. The overly potent buyers drive up the price by spending ridiculous amounts of taxpayer's money. However, there is another factor: Zero-day exploits are very volatile. They "die", resulting in intelligence services having a continuous need for them. And it gets worse: A recent Rand study [24] shows that while an exploit's life expectancy is about seven years, more than 25 percent don't survive the first year. Less than 25 percent survive up to nine years. However, no indicator explains why and which exploit is likely to survive longer than others. "For a given stockpile of zero-day vulnerabilities, after a year approximately 5.7 percent have been discovered and disclosed by others." This is very im-

portant for military usage, where attackers want to be the only ones. The Rand study found that it usually takes 22 days to create a fully functional exploit once a vulnerability has been discovered.

Code Quality and Bug Bounties

Bug bounties are another reason for the high prices of exploits: Companies like Google pay five-digit sums for bugs, to a total of \$3 million in 2017, helping to create a valid business model for hackers – they don't need to actually exploit the flaws to make a living [25].

Many experts claim that the number of available zero-day exploits are decreasing (causing prices to explode), because software vendors' code quality is continuously increasing, resulting in fewer exploitable flaws. Code fuzzing, automated QA, and extended, faster testing are often mentioned as very successful. A former hacker, now trainer and security consultant for German prosecutors, says tools like Google's ClusterFuzz discover bugs before they can be exploited.

The Hacker's Business Model

The hacker's business may have become a lot harder. A former hacker noted that his "last full

weaponized exploit took eight months to create; today you need a full chain from remote to kernel, and not every stack overflow is exploitable anymore – it's not like in the 90s. In '98, all we needed to create an exploit was a crate of Mate, a weekend, some computers, and no sleep." Technologies like address space layout randomization (ASLR), code fuzzing, and the extended use of canaries make it harder today to exploit software bugs, he further explains. "Someone at Pwn2Own had to chain 17 bugs to finally get code execution." Even though the prize money awarded at Pwn2Own 2018 [26] decreased, the work quality increased in skill level. Today, exploit programmers have to know more about operating systems, platforms, and software and invest more time and qualification in their work. Very likely that is also a reason for price increases.

If not for all the taxpayer money being wasted, the whole development could be seen as very positive: "If I find a bug, I can choose to get the \$10 - 20K from Google – not too bad for a few days, maybe weeks of work – or invest months just to find the bug being closed or discovered by someone else in the meantime." With more and more open source, this scenario becomes more and more likely. It seems to be a business decision for a hacker as well.

Ethics

So what about the malware industry? On the one hand, they have enough money to have lots of skilled developers code exploits, even if it takes longer. The large amount of money spent should even make up for occasional losses due to the "death" of a bug before an exploit is finished.

When you talk to these companies' representatives, they usually don't want to talk about this. However, there's one topic they do like to talk about: lack of experts. Both the military and its affiliated businesses can't find experts with skill levels as high as they need. In addition, the ethical differences between the hacker culture and the military's goals pose another problem. Ethical hacking is a big thing; there are even certifications available now. In the meantime, Western military leaders and politicians alike wonder how the Russians motivate their hackers. Although patriotism might have some influence with Russian hackers, ideology is rare – most are more interested in cleaning out other people's bank accounts [27]. ■■■



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

Info

- [1] Tim Cook on Google and Facebook: <http://time.com/5433499/tim-cook-apple-data-privacy/>
- [2] Eisenhower's farewell address: https://en.wikipedia.org/wiki/Eisenhower%27s_farewell_address
- [3] Data is the new oil: <https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data>
- [4] German IT Security 2017: https://www.bsi.bund.de/EN/Publications/SecuritySituation/SecuritySituation_node.html
- [5] Windows zero-day exploit: <https://thehackernews.com/2018/10/windows-zero-day-exploit.html>
- [6] Bleedingbit: <https://www.zdnet.com/article/new-bleedingbit-zero-day-vulnerabilities-impact-majority-of-enterprises-at-the-chip-level/>
- [7] Motherboard article: https://motherboard.vice.com/en_us/article/neqkgm/israel-zero-days-letter-to-american-hackers
- [8] Israeli Ministry of Defense letter: <https://www.documentcloud.org/documents/4389584-Israel-MoD-Zero-Days-Letter.html>
- [9] VEP: https://en.wikipedia.org/wiki/Vulnerabilities_Equities_Process
- [10] EFF VEP FOIA request: https://www.eff.org/files/2016/01/18/37-3_vep_2016.pdf
- [11] New US policy regarding VEP: https://www.theregister.co.uk/2017/11/15/us_governments_vulnerability_disclosure_policy/
- [12] "Old Trick Threatens the Newest Weapons" by John Markoff, *The New York Times*, October 26, 2009: <https://www.nytimes.com/2009/10/27/science/27trojan.html>
- [13] Stuxnet: <https://www.businessinsider.com/stuxnet-was-far-more-dangerous-than-previous-thought-2013-11>
- [14] US cyberwarfare: https://en.wikipedia.org/wiki/Cyberwarfare_in_the_United_States
- [15] National Cyber Strategy: <https://www.whitehouse.gov/wp-content/uploads/2018/09/National-Cyber-Strategy.pdf>
- [16] USCYBERCOM: <https://www.cybercom.mil/About/Mission-and-Vision/>
- [17] Zero Days: <http://www.zerodaysfilm.com/>
- [18] Stuxnet II: <https://www.bleepingcomputer.com/news/security/new-stuxnet-variant-allegedly-struck-iran/>
- [19] DoD Cyber Strategy 2018: https://media.defense.gov/2018/Sep/18/2002041658/-1/-1/1/CYBER_STRATEGY_SUMMARY_FINAL.PDF
- [20] Kurz, Constanze, and Frank Rieger. *Cyberwar – Die Gefahr aus dem Netz*, C. Bertelsmann Verlag, 2018: <https://netzpolitik.org/2018/cyberwar-der-endlose-krieg/> [in German]
- [21] FinFisher: <https://en.wikipedia.org/wiki/FinFisher>
- [22] Adriel Desautels at BDF 2015: <https://www.youtube.com/watch?v=VkembqbnNUQ>
- [23] Zerodium: <https://zerodium.com>
- [24] Rand exploit study: https://www.rand.org/content/dam/rand/pubs/research_reports/RR1700/RR1751/RAND_RR1751.pdf
- [25] Google's bug bounty program: <https://techcrunch.com/2018/02/07/googles-bug-bounty-programs-paid-out-almost-3m-n-2017/>
- [26] Pwn2Own 2018: <https://securityaffairs.co/wordpress/70358/hacking/pwn2own-2018.html>
- [27] Russian hackers: <https://www.calvertjournal.com/features/show/3781/>

FREE DVD

ubuntu®

ECLIPSE TOOLS for IoT projects

INTERNET OF THINGS Tools for home automation

LINUX MAGAZINE

Brand New! Ubuntu 16.04 DVD inside!

INTERNET OF THINGS

Tools for home automation

Bluetooth Tricks Track movements within your house

Linux on Old Hardware

Tablets on Linux Configure graphics settings with xsetwacom

Raspberry Pi 3 B+ Does the world need a new Rasp Pi?

Python Gambling Simulating games of chance

Logism Design and draw digital circuits

MYCROFT An open source personal assistant

PIRATEBOX Handy anonymous file server for parties and meetups

ArchLinux

Double-Disc DVD INSIDE!

WWW.LINUX.MAGAZINE

FREE DVD

ubuntu®

INTERNET OF THINGS Tools for home automation

LINUX MAGAZINE

Brand New! Ubuntu 16.04 DVD inside!

MYCROFT

Can an open source personal assistant compete with Alexa?

Scanners and Linux

Peppermint OS Peppy distro built for the cloud

Meltdown & Spectre Kernel developers take swift action to protect Linux

Knight's Tour Classic chess puzzle gets a Python twist

Node-RED Control a Rasp Pi using text messages

FOSSPicks

- Krita 4.0
- FreeTube
- OpenSnitch
- Yoda

LIBREBOOT and the liberation of firmware

FOSSPicks

- Gimp 2.10
- comix
- Fast Finder
- WireGuard

DOUBLE-DISC DVD INSIDE!

WWW.LINUX.MAGAZINE

FREE DVD

ubuntu®

INTERNET OF THINGS Tools for home automation

LINUX MAGAZINE

Brand New! Ubuntu 16.04 DVD inside!

INSIDE A COMPILER

What really happens to that code

What's New in LibreOffice 5.4

Brown Dog Gadgets More fun with science

Neural Network Can a program play the Monty Hall game?

Catch a Fox! Monitor wildlife traps with a microcontroller

GUI Firewall Tools Airtight security in an easy view

Guetzli JPEG-quality images with a tiny file size

FOSSPicks

- Krita 4.0
- FreeTube
- OpenSnitch
- Yoda

LIBREBOOT and the liberation of firmware

FOSSPicks

- Gimp 2.10
- comix
- Fast Finder
- WireGuard

DOUBLE-DISC DVD INSIDE!

WWW.LINUX.MAGAZINE

FREE DVD

ubuntu®

INTERNET OF THINGS Tools for home automation

LINUX MAGAZINE

Brand New! Ubuntu 16.04 DVD inside!

INSIDE A COMPILER

What really happens to that code

What's New in LibreOffice 5.4

Brown Dog Gadgets More fun with science

Neural Network Can a program play the Monty Hall game?

Catch a Fox! Monitor wildlife traps with a microcontroller

GUI Firewall Tools Airtight security in an easy view

Guetzli JPEG-quality images with a tiny file size

FOSSPicks

- Krita 4.0
- FreeTube
- OpenSnitch
- Yoda

LIBREBOOT and the liberation of firmware

FOSSPicks

- Gimp 2.10
- comix
- Fast Finder
- WireGuard

DOUBLE-DISC DVD INSIDE!

WWW.LINUX.MAGAZINE

FREE DVD

ubuntu®

INTERNET OF THINGS Tools for home automation

LINUX MAGAZINE

Brand New! Ubuntu 16.04 DVD inside!

MAPPING TOOLS

Create and edit digital maps with QGIS and QMapShack

4 Open Source Microblogging Tools

MakerSpace Handicapped Pi projects

Kubuntu Up Close Explore the new 18.04 release

Caddy HTTPS server out of the box

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

Ogg Vorbis Free format for audio files

A Python Script That solves the Chinese river puzzle

FOSSPicks

- Krita 4.0
- FreeTube
- OpenSnitch
- Yoda

LIBREBOOT and the liberation of firmware

FOSSPicks

- Gimp 2.10
- comix
- Fast Finder
- WireGuard

DOUBLE-DISC DVD INSIDE!

WWW.LINUX.MAGAZINE

FREE DVD

ubuntu®

INTERNET OF THINGS Tools for home automation

LINUX MAGAZINE

Brand New! Ubuntu 16.04 DVD inside!

FREE BOOT!

boot and the liberation of firmware

4 COLLABORATIVE OFFICE SUITES: Keep your team in sync

LIBREBOOT and the liberation of firmware

FOSSPicks

- Krita 4.0
- FreeTube
- OpenSnitch
- Yoda

LIBREBOOT and the liberation of firmware

FOSSPicks

- Gimp 2.10
- comix
- Fast Finder
- WireGuard

DOUBLE-DISC DVD INSIDE!

WWW.LINUX.MAGAZINE

FREE DVD

ubuntu®

INTERNET OF THINGS Tools for home automation

LINUX MAGAZINE

Brand New! Ubuntu 16.04 DVD inside!

TERMINAL TUNING

Tools for a Bash shell

What's New in Fedora 27

Do Managers stay productive with a task-management tool

Postal server for a cloud

Feko This new phone puts your privacy first

Rasp Pi Tutorial

HUGE SAVINGS! \$29.99 VALUE! ON A SEARCHABLE DISC!

FOSSPicks

- Krita 4.0
- FreeTube
- OpenSnitch
- Yoda

LIBREBOOT and the liberation of firmware

FOSSPicks

- Gimp 2.10
- comix
- Fast Finder
- WireGuard

DOUBLE-DISC DVD INSIDE!

WWW.LINUX.MAGAZINE

FREE DVD

ubuntu®

INTERNET OF THINGS Tools for home automation

LINUX MAGAZINE

Brand New! Ubuntu 16.04 DVD inside!

SAFER BOOT

Keeping control of the startup process

Font Tricks From the con of your office

Write a script that changes your YouTube metadata

USER! Tricks Add a custom app that runs from the firmware

5 Automated Backup Tools

FOSSPicks

- Krita 4.0
- FreeTube
- OpenSnitch
- Yoda

LIBREBOOT and the liberation of firmware

FOSSPicks

- Gimp 2.10
- comix
- Fast Finder
- WireGuard

DOUBLE-DISC DVD INSIDE!

WWW.LINUX.MAGAZINE

FREE DVD

ubuntu®

INTERNET OF THINGS Tools for home automation

LINUX MAGAZINE

Brand New! Ubuntu 16.04 DVD inside!

SAFER BOOT

Keeping control of the startup process

Font Tricks From the con of your office

Write a script that changes your YouTube metadata

USER! Tricks Add a custom app that runs from the firmware

5 Automated Backup Tools

FOSSPicks

- Krita 4.0
- FreeTube
- OpenSnitch
- Yoda

LIBREBOOT and the liberation of firmware

FOSSPicks

- Gimp 2.10
- comix
- Fast Finder
- WireGuard

DOUBLE-DISC DVD INSIDE!

WWW.LINUX.MAGAZINE

FREE DVD

ubuntu®

INTERNET OF THINGS Tools for home automation

LINUX MAGAZINE

Brand New! Ubuntu 16.04 DVD inside!

Systemd

Getting more from the elusive new Linux init system

10 Top PDF Readers

Arduino Programming for Open Hardware Projects

Homegrown Facial Recognition Build facial recognition into your own Python scripts

Maker Tricks Monitor a beehive with a Raspberry Pi

Cover Your Assets! Back up your system files with CVA

Massive! Arduino explains got started

FOSSPicks

- Krita 4.0
- FreeTube
- OpenSnitch
- Yoda

LIBREBOOT and the liberation of firmware

FOSSPicks

- Gimp 2.10
- comix
- Fast Finder
- WireGuard

DOUBLE-DISC DVD INSIDE!

WWW.LINUX.MAGAZINE

FREE DVD

ubuntu®

INTERNET OF THINGS Tools for home automation

LINUX MAGAZINE

Brand New! Ubuntu 16.04 DVD inside!

PRIVACY

Stop snoopers and protect your identity

Rockbox Liberate your music player with free firmware

RSS Readers View and organize all your favorite news sources

PIXtend More interfaces for your Rasp Pi

Rasp Pi Sailboat Ride the winds with the tiny Pi Zero

Ready to Go? Smart queries in the powerful Go language

FOSSPicks

- Krita 4.0
- FreeTube
- OpenSnitch
- Yoda

LIBREBOOT and the liberation of firmware

FOSSPicks

- Gimp 2.10
- comix
- Fast Finder
- WireGuard

DOUBLE-DISC DVD INSIDE!

WWW.LINUX.MAGAZINE

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



FOSSPicks

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



In between finding the hottest FOSS Picks, Graham has rebuilt his BrewPi and is now brewing a strong winter ale. Hic. **BY GRAHAM MORRISON**

Diagramming

draw.io

Draw.io has been around for years as a website. If you ever need to draw a diagram or flowchart, or even a circuit, it's indispensable. Many users now rely on it for their last-minute presentations, their documentation projects, and project planning, because draw.io's great strength is that it offers many different symbols in many different categories, making it capable of drawing many different kinds of charts. You don't have to create your own icons or steal images from the Internet or rely on Inkscape for your arrowheads. Instead, you simply drag and

drop symbols, join them together, and then use draw.io's wonderful styling, arrangement, and node options to create a final file that you can save online or locally. The only downside is that the website version requires running it online.

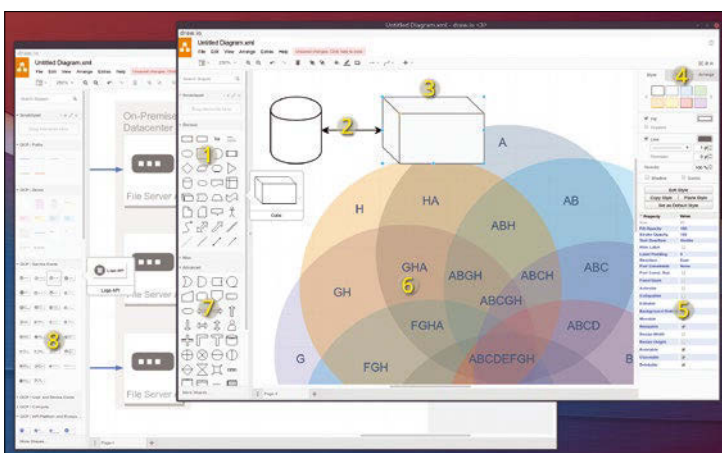
But draw.io is an open source project, and there's now a desktop version you can run on your own local computer, which is perfect for last-minute cramming on that train with abysmal connectivity. There's an executable ApplImage, as well as DEB, RPM, and even Chrome OS files. With the application installed,

you can create diagrams without the online tether. There's a huge number of symbols to choose from, with additional custom symbols embedded within many of the template New projects you can open. The Google Cloud Platform (GCP) templates, for example, include many GCP-centric symbols and layers for use within your charts. Further examples include Venn diagrams, mind maps, Unified Modeling Language (UML) diagrams, flow charts, and isometric network diagrams – with 3D symbols for Amazon Web Services (AWS) infrastructure.

Although not as powerful as Inkscape for drawing, text, or arrangement options, you have more than enough tools here to create the perfect diagram, as well as group, layer, and align, for instance. If you need more control, you can save as an SVG and edit within Inkscape. I tested this, and it worked perfectly. There's even beta support for VSDX output if you need to import your file into Microsoft's Visio. The UI still feels web-based, and native desktop widgets would be preferable, but this is an important application with few desktop rivals – I can't think of any Linux tools that are able to generate diagrams of this quality so quickly, and none with the same powerful design and symbol library that can output in so many different formats. Diagrams aren't necessarily the most exciting thing to spend your time creating, which is precisely why draw.io is so good. It handles the tedious parts, so you can spend less time drawing and more time moving on to the implementation (or at least arguing over the implementation shown in your awesome diagrams).

Project Website

<https://github.com/jgraph/drawio>



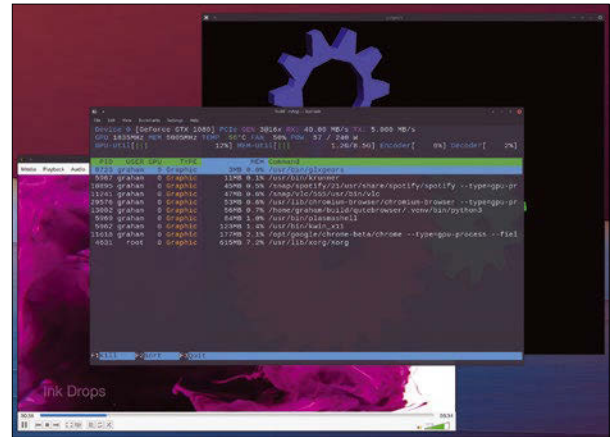
- 1 Symbol library:** Drag and drop from a huge variety of diagram icons.
- 2 Connections:** Components can be dragged, and their connections remain intact.
- 3 Edit:** Scale, drag, rotate, and group elements on your canvas. **4 Style:** Use a color palette for fill styles, plus control fonts and labels separately. **5 Properties:** If the mouse isn't good enough, type in exact values. **6 Transparency:** Make and mix your colors with opacity values. **7 Shape categories:** Symbols cover a huge range of diagram types. **8 Custom libraries:** Include your own symbols, such as GCP or AWS shapes, to accompany your own diagrams.

GPU monitor

nvtop

Despite the NVidia graphics card being an essential part of many Linux setups, especially if you're into virtual reality (VR), gaming, or machine learning, there isn't much in the way of third-party (or first-party!) support. NVidia X Server Settings is about as good as it gets. With the proprietary drivers installed, the hardware is often left simply to find its own way, whether that's accelerating polygons with OpenGL or autonomous truth matrices with OpenCL. The first sign of anything going awry is likely to be the airplane sound of your GPU fans as they attempt to cool some wayward GPU-bound process. If the GPU was a CPU, you'd use a tool like `top`, or `htop`. Now with NVidia's hardware, you can use `nvtop`.

`nvtop` is a task monitor for your graphics hardware. Much like the CPU variants, it shows you the type of application, the amount and percentage of memory it's using, and the command used to launch the process. A global section also displays details about your specific hardware, such as the model, PCI speed, GPU frequency, transfer rates, power, temperature, and fan. A couple of brilliant charts show percentage usage for raw processing power, memory, and encoding and decoding, as well. Many of these elements are dynamic and will change as your GPU experiences periods of load. It provides a fascinating insight into something not often considered, but if you're using your GPU for work or seri-



If you've ever wondered just how much of your GPU the plasmashell uses, wonder no longer.

ous gaming, it's also essential because you can see which processes are stealing precious cycles and affecting your performance. You can then kill them directly from within the text-based UI. It's all you need for GPU management. If you have an NVidia GPU, this is a tool that will quickly become an essential build and install on any system on which you rely.

Project Website

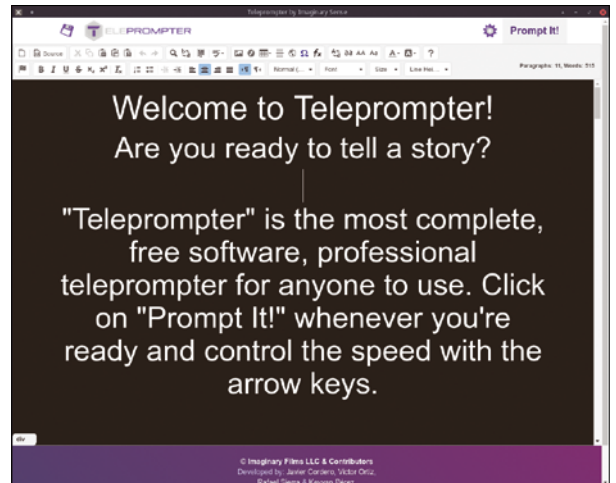
<https://github.com/Syllo/nvtop>

Teleprompter

Imaginary Teleprompter

A teleprompter is often associated with that awkward wide-angle shot of live television, where the presenter is looking past the camera towards the large screen of text showing what to say. The teleprompter is usually showing a script for the presenter to speak, and it's been an important but specialized aspect of television production for decades. There initially seems little reason to look at an open source version, but YouTube has made all those old television techniques new again – from lighting to blue screens – and a teleprompter is a great way to help with narration and flow and takes the pressure off the presenter needing to remember everything to say.

Imaginary Teleprompter is one such teleprompter, and it looks and operates just like a word processor. You edit and format the text in exactly the same way, only you typically make the text very large. The main window lets you add images, create anchor points, add tables, and even add flash animations. A real-time update shows the number of paragraphs you've added, alongside the number of words. When you're happy with your copy, you click *Prompt it*, which opens a new pane that gives you the option to flip the screen, use an external prompter, change the style from the standard blackboard, and set the focus area. The focus area is a narrow letterbox through which the text scrolls – like a window – allow-



The only feature we'd love to see in Imaginary Teleprompter is an estimate of how long a talk is going to take.

ing the presenter to see what's coming and can then get ahead of the focus point or behind, depending on the presenter's style and the context of the talk. As you'd expect, you can also change the scroll speed, as well as the font scale and acceleration curve. A timer also shows you how long you're taking. It all works absolutely brilliantly and could easily make the difference between your YouTube channel being full of clumsy reviews and a professional channel that's far more likely to attract viewers and sponsors.

Project Website

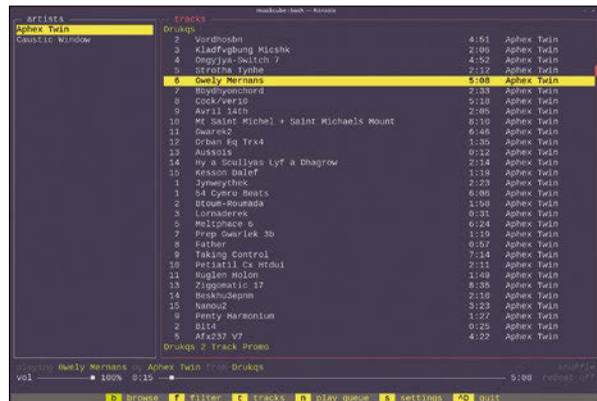
<https://snapcraft.io/imaginary-teleprompter>

Music player

musikcube

It's been said before: Rather than the command line becoming extinct as desktop GUIs take over, it seems the opposite is happening. Each month, an increasing number of wonderful command-line interface (CLI) tools attempt to do things we've become more accustomed to on the desktop. Of course, this could be because it's easier to create a command-line tool than a desktop tool, and it's a good place to start if you're experimenting with Linux and open source software development. However, this can't be the whole story. It must also be because more and more of us are beginning to prefer the distraction-free environment of the command line over the desktop. This is certainly my own experience.

Musikcube is another step away from the desktop without having to sacrifice any desktop functionality. It's a music player and manager alongside a server that can stream your music to other clients and even an Android app. It works just like any major music application, building a database of your music collection and then letting you browse by album or artist. The filter section lets you quickly search for a piece of music, an album, or an artist that matches your search term. You simply add the music you're interested in to the play queue. It's easy to use, with your first view providing a great overview of how musikcube is configured and which music back end it's using (PulseAudio by default). You just



Although not shown here, the Android client is a brilliant accompaniment to an already brilliant application.

need to select where your music is stored and press the space bar. From there, you simply navigate through your music and press play. The UI is beautifully designed and looks fantastic on a dark theme, making a mockery of the frippery you find in some desktop equivalents.

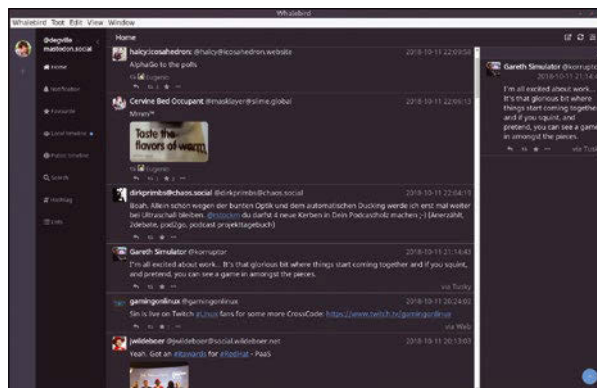
Project Website
<https://github.com/clangen/musikcube>

Mastodon client

Whalebird

There's no doubt that the popular social networks are going through something of an identity crisis. After their incredible first flushes of success, we're now at the stage where each is attempting to heavily monetize its user base. This is rightfully waking many users up to how their data is being used. Federated social networks are often touted as the answer, because there's no single gatekeeper to your posts and no one with control over the whole network. But to be useful, a federated network needs to get past the chicken and egg situation of becoming useful before more people join. Mastodon is the current popular federated option, and it does seem to have momentum, at least

with the open source and privacy communities. Of the approximate 750 people I follow on Twitter, I found 35 on Mastodon. That's only around 5 percent of an already self-selected bias for this kind of technology, but at least I can finally keep on top of my timeline. To make all this useful, you need a decent client. Mastodon's web interface has improved hugely recently and will feel familiar to anyone who uses TweetDeck for Twitter. It defaults to a multicolumn view that's most useful to the power user who needs to use many timeline filters. There are several decent command-line clients, too. However, Whalebird, an Electron desktop application, is a great choice if you're just starting out.



One of Mastodon's best features is that you can now automatically bridge it to your Twitter account and follow the same people on Mastodon automatically.

Like the standard Twitter experience, Whalebird doesn't overwhelm you with options and focuses on your own basic timeline and any community interaction. It looks a lot like the Slack client, for instance, and features decent shortcuts, a dark theme, hashtag views, and lists. If you've been on the fence about using yet another social network, Whalebird could just make the difference.

Project Website
<https://whalebird.org>

Repository viewer

GRV

So many of us now use `git` on the command line, it's difficult to contemplate using a GUI tool to make accessing Git repositories easier. The context shift from command line to desktop is often enough to break your train of thought or concentration, but we have covered one helpful command that's worth using, `tig`, and here's another. GRV (Git Repository Viewer) is a lot more ambitious and effectively builds an entire GUI on the command line around your local Git repositories. The main view, called the History View, consists of three panels. On the left is a pane for branches, tags, and remote references, and in the middle is a list of commits for your chosen branch. The pane on the right side lists the changes, as

a `diff`, for that specific commit. The other view is Status View, which gives an overview of what's changed in your local branch since the last push. If you're already familiar with `git`, this makes complete sense. It's wonderful to see your work presented in such a clear way, just as you likely imagine the whole repository hanging somewhere.

GRV is also deep. It defaults to using Vim key bindings and makes a wonderful partner to programming in the editor. There's even theme support and the ability to filter through the data using a query language rather than simple searches. In the background, changes to the repository are captured by monitoring the file-system and update the UI automatically. This means it works

Make sense of your own git repos, and those of other developers, with the wonderful GRV.

well in a separate `tmux` panel, for example, capturing changes as you make them, or as they're pulled from upstream. Git is complicated; while GRV is never going to make the learning curve any easier, it's a fantastic tool for confirming what you think is happening is actually happening, as well as helping you to visualize how your projects are developing.

Project Website

<https://github.com/rgburke/grv>

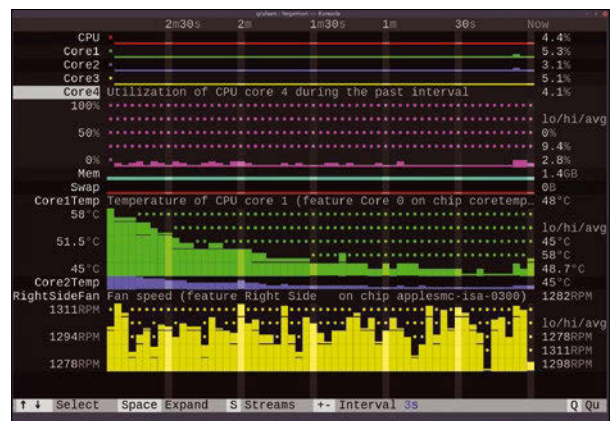
System monitor

Hegemon

There are many command-line system monitor tools, and we've looked at a fair number in these pages, but there's always room for another monitor that attempts to do something differently. Hegemon succeeds, mostly, because it presents only the information we really care about in a way that's useful. That information is how much of your CPU capacity is being used (broken down into cores, if required), how hot your system is running, and how well your fan or cooling is responding to this load. It presents this information as a scrolling and colorful histogram that can be expanded or contracted with a simple press of the space bar. As system load increases, you

can see exactly which cores are carrying the burden, along with how it's heating up your system. The fan speed will typically ramp up to compensate, creating a scrolling chart of your system load and on-going performance.

The folding and unfolding of each element is a quick and easy way to limit the details you see, as well as to get more information when you need it. To Hegemon, each of these sources is a data stream, effectively making the application modular, and further data streams can potentially be plugged into the same system. The excellent documentation covers the details and also promises incoming network, disk I/O, and GPU usage streams to augment the already busy display. The application it-



We need a new category for command-line tools like Hegemon that would look amazing in hacker-related films.

self is written in Rust and is installed via Cargo, which makes it a little more install-resource-heavy than the typical command-driven monitor. However, this means writing a new data stream could be a perfect point of entry for someone's nascent Rust skills and a great way of adding to an already fully functional and useful tool.

Project Website

<https://github.com/p-e-w/hegemon>

Text editor

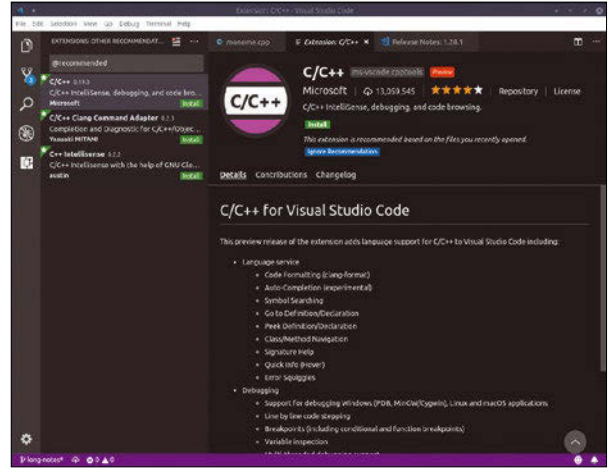
Visual Studio Code

Microsoft has changed its strategy towards both Linux and open source, no doubt because it realizes that many developers find working within a Linux environment indispensable and because cloud-based Linux deployments dominate, even on its own Azure platform. But it's still surprising to find Microsoft joining The Linux Foundation and, more recently, the Open Inventions Network. At its inauguration at the latter, Microsoft pledged that 60,000 of its patents won't be used against other members. Of course, these choices are likely to be pure business decisions, and there's absolutely nothing wrong with that. It proves that the open source approach we've all been pursuing for decades is, pragmatically, the best approach for innovation.

Outside of Microsoft's business decisions, though, it's also been releasing its own open source software. If you use .NET, for example, you'll find having Linux access to Power-

Shell immensely useful, as you'll no longer need to dual-boot or boot into Windows. Visual Studio Code is another Microsoft convenience feature you're going to find useful if you spend much time with Windows. Sadly, it's not the fully fledged IDE that "Visual Studio" implies, and no doubt many developers would love to see Visual Studio running on Linux. But this text editor is the next best thing.

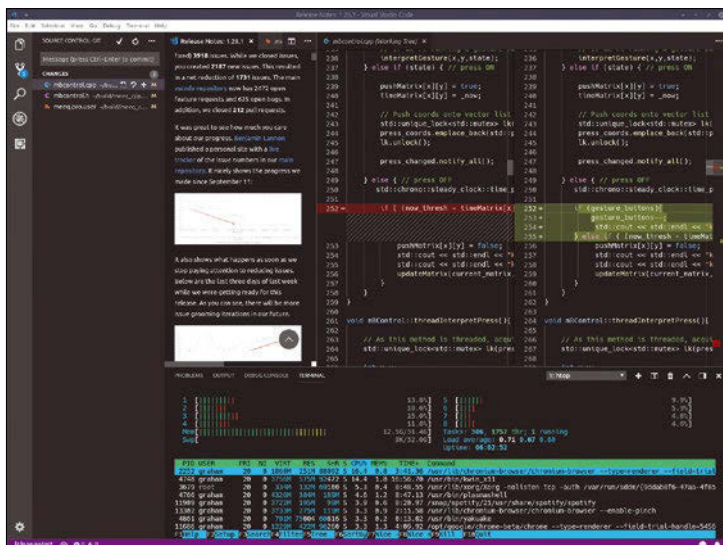
First, a caveat: Don't download the Microsoft-provided binary, bundled even in the Arch AUR package. The binary includes Microsoft's original telemetry and will send back usage statistics to Microsoft. Instead, you can grab a binary from VSCode's GitHub repository, a fork with all of this removed, or you can build it yourself. If you spend any time with code, it's worth the trouble, because Visual Studio Code is an excellent editor with a few unique features of its own. In particular, it has excellent support for many different languages and formats, including C,



Some excellent extensions can be installed to augment the editor for your language and environment.

C#, C++, Clojure, Python, Swift, and Markdown. These top-tier languages can take advantage of snippets, syntax highlighting, brace matching, and code folding. But one of the best things about the editor are the additional extensions that can be installed. If you're using one of these languages, or one of the many others that are supported, the editor will even recommend which to install. C++ users are prompted to install IntelliSense, for example, which is Microsoft's rather good code formatting, code completion, method navigation, and debugging plugin.

The source control pane is also excellent when used with a Git repository. Its diff view for showing changes made in the current branch is one of the best I've used, and you can commit those changes in-place without leaving the editor. Thanks to Electron, font rendering is beautiful, too. You have several excellent themes to choose from, and you can easily build your own. The tabbed interface is quick and easy to navigate, and any view can be split and moved around. There's a release version, which means development is moving quickly. Although it's not Visual Studio running on Linux, Visual Studio Code is a great alternative that favorably competes with some of the best native editors we've all been using for years.



Visual Studio Code is a great editor with brilliant debugging abilities, multiple splits, and an embedded command line.

Project Website
<https://github.com/VSCodium/vscodium>

NES emulator

ANESE

When it comes to games consoles, the Nintendo Entertainment System (NES), is an unassailable classic. For a certain generation, it's the only games console that has ever been, and Nintendo is still milking many of the characters and game mechanics that first saw light on the NES over three decades ago. As you'd expect for an old, popular games console built around a humble 6502 8-bit CPU, the NES has been successfully emulated for a long time, and any new emulator is going to need to offer something new to get noticed. ANESE does exactly this. ANESE is a great little project that's built using "clean and interesting" C++11 with an emphasis on readability, maintainability, and approachability. The code has plenty

of comments, and the entire project has very few dependencies, making it easy to build. If you've ever wondered how emulation works or how to get into it yourself, this is a great project to study.

But ANESE also offers a new and unique feature that any NES aficionado is going to find fascinating. This is a mode that's enabled by launching the emulator with a `--widenes` argument, which, when enabled, will allow the emulator window to grow outside of the screen area as you explore the game level of whatever you're playing. As you move right through Mario World, for example, you'll begin to see the entire world mapped out for you. In this way, you can create maps of entire games as you play through them in a way that isn't easily possible



ANESE is a great new NES emulator with its own wonderful unique feature.

by simply extracting the data from the cartridge. It gives you a brilliant insight into how the games were designed and can obviously be used to cheat or find shortcuts if you're looking for speedruns. But mostly, it just leaves you appreciating the wonderful games design of the time.

Project Website
<https://prilik.com/ANESE/>

Crash simulator

Rigs of Rods

If you've ever played a car racing game and been disappointed by the totally unrealistic collision mechanics when your car hits an obstacle, this game is for you. Like Destruction Derby, Rigs of Rods' entire premise is to smash things up, although there's no scoreboard or reward for doing so. Instead, this is a physics sandbox created for experimenting with vehicles and damage. Load up a terrain, add a few cars, and start driving them around. The driving mechanics are actually very good. You see in-car controls and dials, as well as status indicators for all the main components. You can change your view from first to third person. As some of the terrains feature roads and cities, you can slide your SUVs or

sports cars around like a maniac. When you eventually do hit something, the physics are satisfyingly realistic, especially in the way the object meshes fall apart and deform. It's exactly this kind of deconstruction that's missing from nearly all driving games.

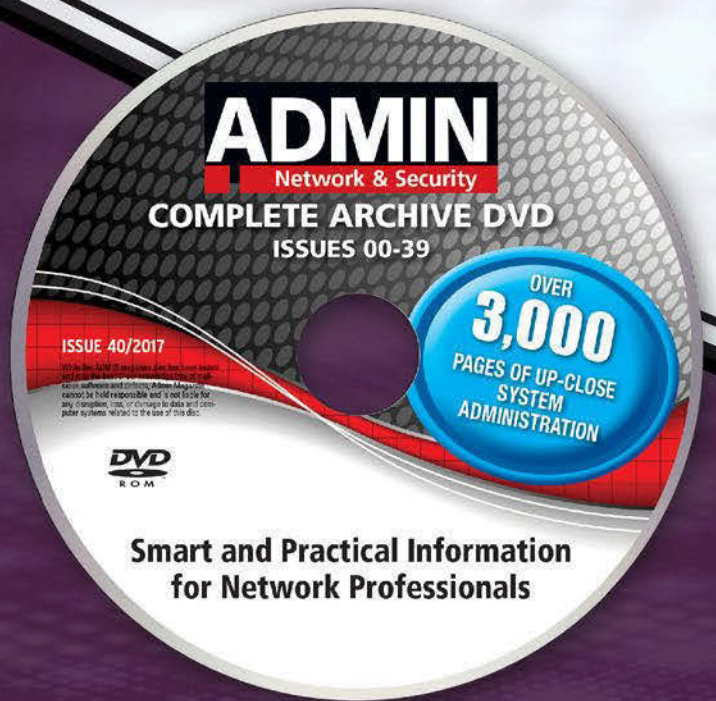
Almost all of the content is created by the community, with hundreds of cars, textures, and terrains for download and experimentation. You can create your own content using a tool like Blender and the AngelScript programming language to add some logic to your creations. There's even a multiplayer mode for playing with your friends; up to 64 users can join you to create a scene of utter mayhem. You can even click and drag objects around with



Rigs of Rods is a driving game where you're actually rewarded for bad driving.

your mouse, including physical effects, effectively turning Rigs of Rods into a god game. It's a lot of fun and feels somewhat reminiscent of Stunt Car Racer as you attempt to hold everything together to turn the car around for one more tilt of the lance at your opponent.

Project Website
<https://www.rigsofrods.org/>



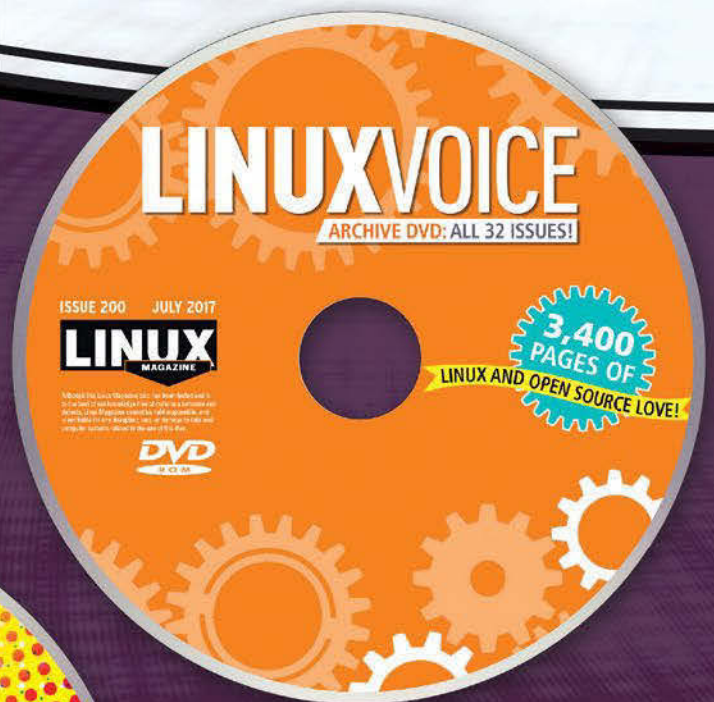
Complete Your Open Source Library with Archive DVDs!

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



Pretty (Inter)face

If you want features, bells and whistles, and configurability in spades, your best choice of desktop is probably KDE's Plasma desktop. Navigating and discovering all that's on offer can be a challenge, though.

BY PAUL BROWN

While many user interface designers advocate simplicity and simplified decision-making for users (which often results in no decision-making at all), the KDE community [1] has stubbornly gone the other way, jam-packing all manner of features and doodads into its Plasma [2] desktop (see the "KDE Is Not a Desktop" box).

That said, if you want simple, Plasma can do simple, too. You can ignore all the bell and whistles and just get on with your life. But where is the fun in that?

Camouflage

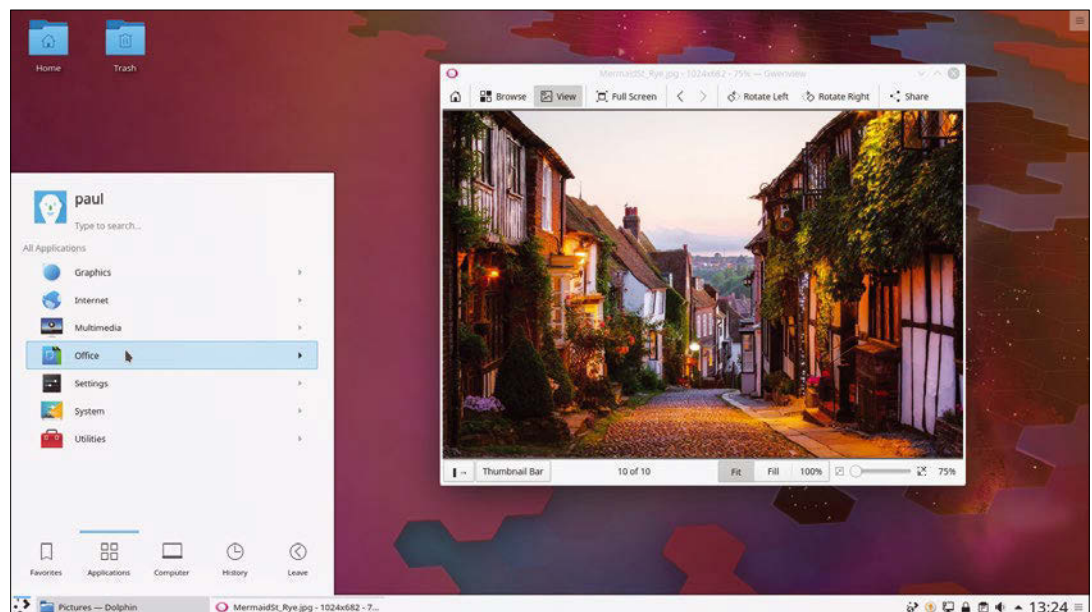
A default Plasma desktop looks like Figure 1. Usually, you will find a panel at the bottom of the screen, a start button holding menus at the bottom left, and a tray on the right – all quite conventional, boring, and even Windows-y.

But Plasma can be configured to look like anything, even like Ubuntu's defunct Unity (Figure 2), Gnome, Mac OS, or whatever else rocks your boat.

To illustrate Plasma's flexibility, I'll show you some tricks you can use to emulate other desktops, starting with global menus. Both Unity and Mac OS use a global menu: It is the menu that appears in a bar at the top of the screen and shows a selected application's options, instead of having them in a bar along the top of the application.

To create global menus in Plasma, first right-click in any free space on the Plasma desktop and select *+ Add Panel | Empty Panel* from the pop-up menu. Usually, the panel will appear across the top of the screen, because the bottom is already filled with the default Plasma panel. If it has popped up anywhere else, click on the hamburger menu (the button with three

Figure 1: Out of the box, Plasma's default layout looks pretty conventional ...



horizontal lines at one end of the panel) and then click and hold the *Screen Edge* button and drag the panel to the top.

Once you have placed the panel, click on the hamburger menu on the right of the panel again and click on *+ Add Widgets*. A bar with all the available widgets will show up on the screen's left. You can narrow your search down by typing *global* into the search box. When you see the *Global Menu* widget, double-click on it, and it will be added to the panel.

Initially you may not see any difference. Indeed, open applications still have their own menubars. Close and reopen the applications, and you'll see how now their menus have moved from the application window to the upper panel you just made.

To make the effect even more striking, click the Start menu on the bottom panel and pick *System Settings*. Under *Workspace Theme*, choose *Breeze Dark* and click *Apply*. You will end up with something like Figure 3.

But theming is just one of the things you can do to tweak Plasma's look and feel. You can also configure the look, size, and location of individual applications and even individual windows to absurd lengths.

Window Hacking

Right-click on any window's titlebar, and a menu will pop up. Apart from the options to minimize, maximize, and close the windows, you'll notice the *More Actions* option. The *Keep Above Others* and



Figure 2: ... but Plasma can adopt any layout that tickles your fancy. Here, Plasma looks like Ubuntu's defunct Unity desktop.

Keep Below Others options are self-explanatory, but you can also make a window *Fullscreen*, and it will be maximized; the application's titlebar and any other desktop element (like panels) will disappear, giving you maximum workspace. If the application doesn't offer you a way to exit full screen, press *Alt+F3* and use the menu to deactivate full-screen mode.

You can also "shade" the window, which means it will roll up like a blind, leaving only the titlebar visible. Another alternative is to remove the border and titlebar, leaving a bare window with no decorations. To recover borders and the titlebar, select the window and press *Alt+F3* again to open the window's configuration menu.

Within *Window Specific Settings* and *Application Specific Settings*, you have all manner of options to fix the application window's position and size. You can make a window stick to a certain area of your screen and become unmovable. At the same time, you can adjust its size to the pixel. You can configure things so that, when you launch a certain application, it always opens in a certain place, maximized, or shaded. You can make the application so it won't close, or you can choose actions from dozens of other options.

Window Manager Settings opens another cornucopia of options for adjusting windows. From

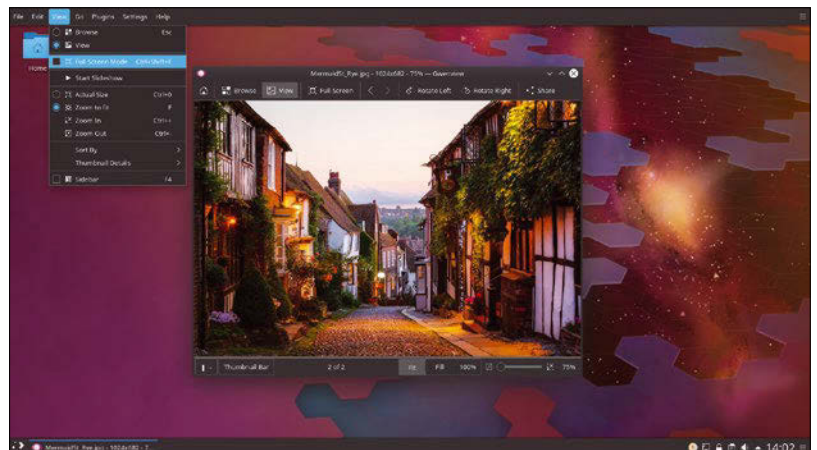
Figure 3: Creating a panel with a global menu is easy.

KDE Is Not a Desktop

This has been the subject of much controversy and confusion, but, no, KDE is not the name of a desktop environment anymore and hasn't been for some time now.

The desktop is called Plasma. KDE, on the other hand, is the name given to the community of developers, artists, translators, and so on that create the software. The reason for this shift is because the KDE community builds many things, like Krita, Kdenlive, digiKam, GCompris, and so on, not just Plasma. Many of these applications are not even tied to Linux, much less to the Plasma desktop, and can be run on many other graphical environments, including Mac OS X, Windows, Android, and others.

Also, much like KFC does not stand for *Kentucky Fried Chicken* anymore, neither does KDE stand for *Kool Desktop Environment*. KDE is not an acronym for anything. It is just ... KDE.



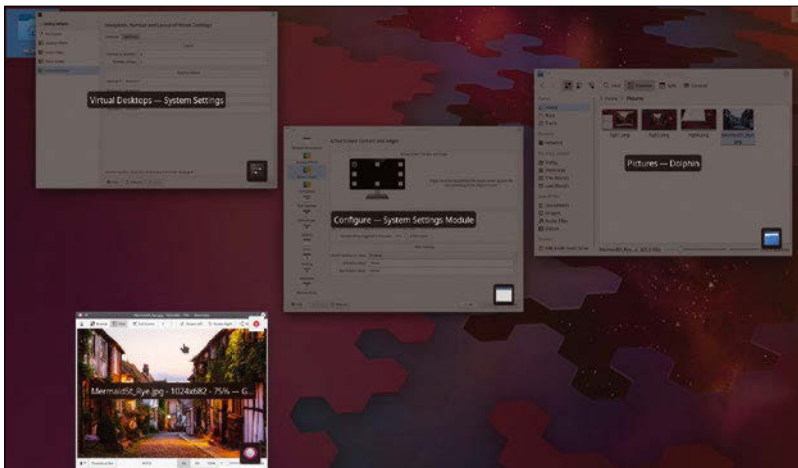


Figure 4: Screen edges allow you to activate actions when you mouse over them.

here, you can adjust things like the screen edges, for example. These are “live” areas that react and carry out an action when your cursor moves into them. Move your cursor into the upper left-hand corner of your screen, and you will see all the windows spread out and go a shade darker, showing everything you have open. You can use this trick to pick a window on which to focus, especially if you have several open (Figure 4).

Figure 5: Rotating cubes are back and can be activated from the Effects tab in Window Manager Settings.

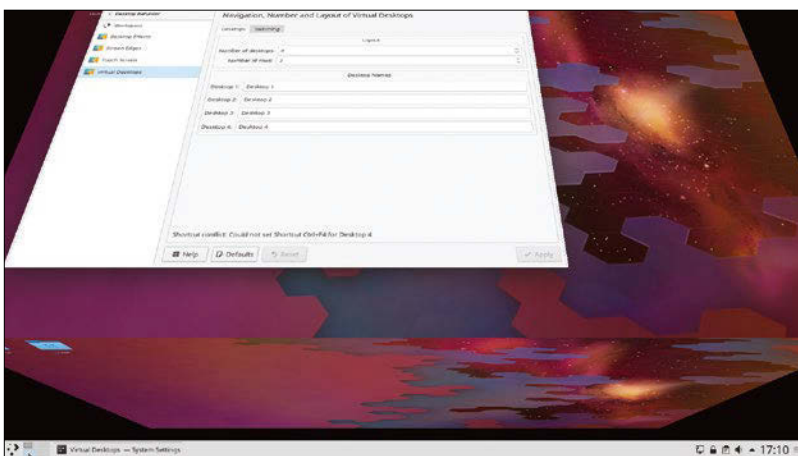
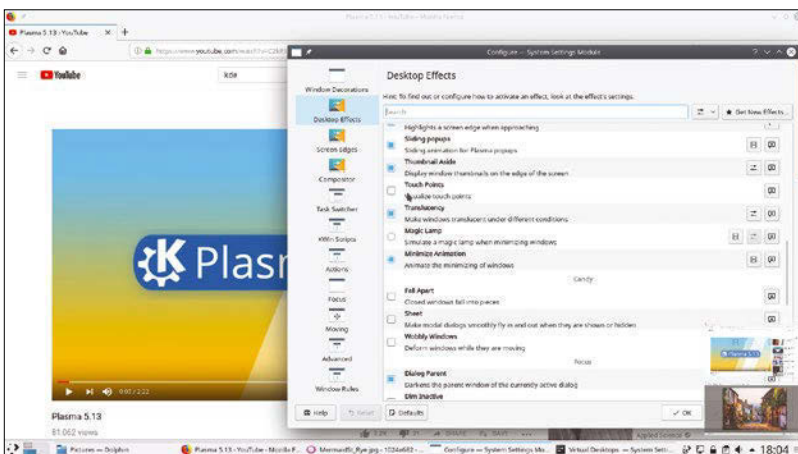


Figure 6: The *Thumbnail Aside* effect allows you to open previews of window activities in the bottom right corner of your screen.



As with everything Plasma, screen edges can be configured to execute a wide variety of other actions: You can minimize all the windows to show the desktop, open KRunner to run a command (you’ll learn more about how KRunner works below), or open the window-switching panel on the left of the screen. The switching panel allows you to scroll through all the open windows to select the one you want.

Another of the more interesting configuration tabs in *Window Manager Settings* is the *Effects* tab. Old timers will remember things like wobbly windows and rotating cubes that plagued Linux desktops of yore. Those still live on in Plasma (Figure 5) and can be activated from this tab, but other effects are more intriguing and worth a second look.

Take, for example, the *Thumbnail Aside* effect. Activate this effect, maximize a window, and press Alt+Meta+T (the Meta key usually has a Windows logo on it), and you will see a small preview of the window appear in the bottom right-hand corner of your screen. You can make these previews translucent (the level of translucency is, of course, adjustable), so you can still see what is under the preview. You can also adjust the size to whatever you want. Click on another window, maximize it, press Alt+Meta+T again, and lo and behold: Another preview will pop up above the first one (Figure 6).

Because these previews do not interfere with the cursor or the other windows, you can carry on with your work while you keep an eye on a video streaming in VLC or a complex install running in a terminal.

Certain caveats apply, though: As mentioned above, the windows you want to preview must be maximized. If you minimize a previewed window, the preview will disappear, but not close, leaving an empty space in the column of previews. Also, as shown above, you can remove borders and the titlebar from windows quite easily, and the *Thumbnail Aside* effect will not work with windows without borders.

Select the previewed window and press Alt+Meta+T again to close its preview.

Being Productive

While on the subject of shortcuts, you always have a terminal close at hand in Plasma: Hit F12 and a Yakuake terminal [3] unfolds from the top of the screen. You can run anything you need from this console. When you’re done, hit F12 again, or click on a window, and the terminal folds back up, moving out of the way. If you are carrying out a long process, like compiling an application, folding the terminal up won’t interrupt it: It will carry on in the background, so at any moment you can hit F12 again and check on its progress.

With regard to productivity/shortcuts/stuff that unfolds from the top of the screen, try pressing Alt+F2 (or Alt+Space, or simply click on an empty area in the desktop and start

typing). A text box for searches appears at the top of the screen. This is KRunner [4], a built-in application originally designed to help search for (and run) applications. Nowadays, KRunner indexes everything: applications, for sure, but also email addresses, packages available from your software center, bookmarks from your web browser, folders, and documents (Figure 7). It is blindingly fast and coughs up results as you type.

KRunner can also be used as a media player (type *play* and the name of the song), a clock (type *time*), a calendar (*date*), a regular and scientific calculator (try typing $=\text{solve}(2*x^2 + 3*x - 2 = 0)$); the first result will be the solution to that polynomial function), as a unit and currency converter (type *100 GBP* to see the value of 100 British pounds in different currencies), and a way to connect to remote computers (type *fish://<your remote server>* to open your SSH server in your file explorer).

KRunner is the epitome of what Plasma is all about.

Misconceptions

You might be thinking that all these goodies are cool and all, but at what cost? Surely Plasma will slow down your average machine to a crawl. Turns out that is not the case at all. In fact, some not-very-scientific research I carried out showed Plasma to be lighter in many areas than even Xfce, a desktop environment whose main claim to fame is that it is light.

Using three Live Manjaro flavors, KDE (with Plasma desktop), Gnome, and Xfce, I ran some tests and concocted Table 1.

If the numbers in the first column seem a bit high, it is because I was running the desktops off of a Live distribution, so, apart from the graphical environment, the RAM was also loaded with a lot of in-memory systems. Notwithstanding, Plasma is the lightest of the three by quite a measure.

Bootup, which was timed from the GRUB screen to a fully loaded desktop, does show Plasma to be slower, but starting up external applications shows Plasma to be faster – in the case of running LibreOffice, much faster. This exercise shows that you can expect Plasma to take longer to load, but, once loaded, it will run lighter and snappier than many other desktops.

Table 1: How Light Is Light?

| Desktop | RAM Used | Bootup | Firefox Startup | LibreOffice Startup |
|---------|----------|--------|-----------------|---------------------|
| Gnome | 542MiB | ~51sec | 7.41sec | 10.75sec |
| Xfce | 530MiB | ~45sec | 7.66sec | 10.10sec |
| Plasma | 489MiB | ~60sec | 7.40sec | 8.07sec |

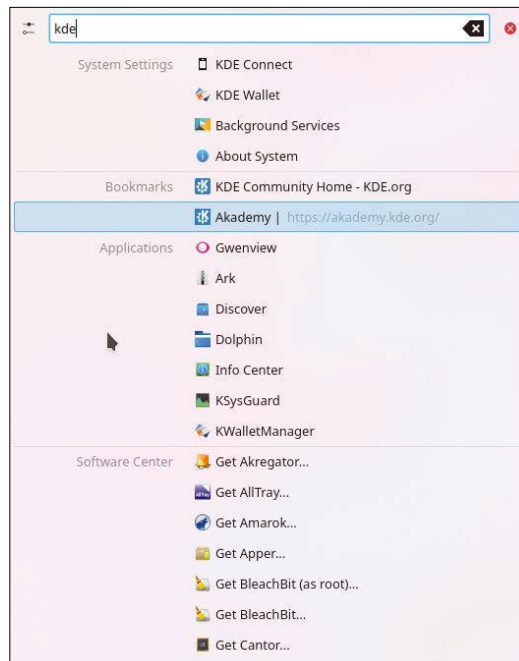


Figure 7: KRunner is your one-stop solution to finding the answer to everything.

The fact is that some manufacturers of underpowered and inexpensive netbooks [5] have started using Plasma by default on their devices.

Conclusion

If you have not tried Plasma or remember the bad old days when it was clunky and buggy, you should really give it a chance. Not only is it configurable to absurd extremes and packs more features than you would ever need, but it is also lighter and snappier than most other desktops out there. ■■■

Info

- [1] KDE: <https://www.kde.org/>
- [2] Plasma: <https://www.kde.org/plasma-desktop>
- [3] Yakuake: <https://www.kde.org/applications/system/yakuake/>
- [4] KRunner: <https://userbase.kde.org/Plasma/Krunner>
- [5] The \$99 Pinebook: https://www.pine64.org/?page_id=3707

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.



Linux of Things

Date: January 21-25, 2019

Location: Christchurch, New Zealand

Website: <https://linux.conf.au/>

Themed Linux of Things, the 2019 linux.conf.au will attract speakers and attendees from across the world. The conference will explore the use of open source software and hardware for internet of things devices, along with security concerns, privacy and legal aspects, environmental impacts, everyday communication, health, ethics, and much more.

FOSDEM '19

Date: February 2-3, 2019

Location: Brussels, Belgium

Website: <https://fosdem.org/2019/>

FOSDEM is a two-day event organized by volunteers to promote the widespread use of free and open source software. FOSDEM is a free event for software developers to meet, share ideas, and collaborate. Every year, thousands of developers of free and open source software from all over the world gather at the event in Brussels.

SCaLE 17x

Date: March 7-10, 2019

Location: Pasadena, California

Website: <https://www.socallinuxexpo.org/scale/17x>

SCaLE – the 17th annual Southern California Linux Expo – is the largest community-run open-source and free software conference in North America. SCaLE 17x expects to host 150 exhibitors this year, along with nearly 130 sessions, tutorials and special events.

Events

| | | | |
|---|----------------------|----------------------------|---|
| IT-Tage 2018 | December 10-13, 2018 | Frankfurt, Germany | https://www.ittage.informatik-aktuell.de/ |
| KubeCon CloudNativeCon North America 2018 | December 10-13, 2018 | Seattle, Washington | https://events.linuxfoundation.org/events/kubecon-cloudnativecon-north-america-2018/ |
| Automotive Grade Linux | January 8-11, 2019 | Las Vegas, Nevada | https://www.automotivelinux.org/event/ces-2019 |
| Linux of Things (linux.conf.au) | January 21-25, 2019 | Christchurch, New Zealand | https://linux.conf.au/ |
| Enigma 2019 | January 28-30, 2019 | Burlingame, California | https://www.usenix.org/conference/enigma2019 |
| Fosdem '19 | February 2-3, 2019 | Brussels, Belgium | https://fosdem.org/2019/ |
| FAST '19 | February 25-28, 2019 | Boston, Massachusetts | https://www.usenix.org/conference/fast19 |
| SCaLE 17x | March 7-10, 2019 | Pasadena, California | https://www.socallinuxexpo.org/scale/17x |
| Open Source Leadership Summit | March 12-14, 2019 | Half Moon Bay, California | https://events.linuxfoundation.org/events/open-source-leadership-summit-2019/ |
| Icinga Camp Berlin | March 14, 2019 | Berlin, Germany | https://www.icinga.com/events/icinga-camp-berlin/ |
| Chemnitzer Linux-Tage | March 16-17, 2019 | Chemnitz, Germany | https://chemnitzer.linux-tage.de/2019/en/ |
| SREcon 19 Americas | March 25-27, 2019 | Brooklyn, New York | https://www.usenix.org/conference/srecon19americas |
| Cloud Foundry North America 2019 | April 2-4, 2019 | Philadelphia, Pennsylvania | https://www.cloudfoundry.org/event/nasummit2019/# |
| Open Networking Summit (ONS) | April 3-5, 2019 | San Jose, California | https://events.linuxfoundation.org/events/open-networking-summit-north-america-2019/ |

CALL FOR PAPERS

We are always looking for good articles on Linux and the tools of the Linux environment. Although we will consider any topic, the following themes are of special interest:

- System administration
- Useful tips and tools
- Security, both news and techniques
- Product reviews, especially from real-world experience
- Community news and projects

If you have an idea, send a proposal with an outline, an estimate of the length, a description of your background, and contact information to edit@linux-magazine.com.



The technical level of the article should be consistent with what you normally read in *Linux Magazine*. Remember that *Linux Magazine* is read in many countries, and your article may be translated into one of our sister publications. Therefore, it is best to avoid using slang and idioms that might not be understood by all readers.

Be careful when referring to dates or events in the future. Many weeks could pass between your manuscript submission and the final copy reaching the reader's hands. When submitting proposals or manuscripts, please use a subject line in your email message that helps us identify your message as an article proposal. Screenshots and other supporting materials are always welcome.

Additional information is available at:

http://www.linux-magazine.com/contact/write_for_us.

NOW PRINTED ON recycled paper from 100% post-consumer waste; no chlorine bleach is used in the production process.

Authors

| | |
|-------------------|--------|
| Erik Bärwaldt | 72 |
| Swapnil Bhartiya | 8, 28 |
| Paul Brown | 92 |
| Zack Brown | 12 |
| Bruce Byfield | 36, 62 |
| Joe Casad | 3 |
| Mark Crutch | 69 |
| Markus Feilner | 52, 76 |
| Karsten Günther | 18 |
| Jon "maddog" Hall | 71 |
| Richard Ibbotson | 16 |
| Roman Jordan | 32 |
| Charly Kühnast | 39 |
| Christoph Langner | 32 |
| Martin Loschwitz | 46 |
| Vincent Mealing | 69 |
| Pete Metcalfe | 66 |
| Martin Mohr | 58 |
| Graham Morrison | 84 |
| Mike Schilli | 42 |

Contact Info

Editor in Chief

Joe Casad, jcasad@linux-magazine.com

Copy Editor

Amy Pettle

News Editor

Swapnil Bhartiya

Editor Emerita Nomadica

Rita L Sooby

Localization & Translation

Ian Travis

Layout

Dena Friesen, Lori White

Cover Design

Dena Friesen

Cover Image

© krisckam, 123RF.com

Advertising

Brian Osborn, bosborn@linuxnewmedia.com
phone +49 89 3090 5128

Marketing Communications

Gwen Clark, gclark@linuxnewmedia.com
Linux New Media USA, LLC
2721 W 6th St, Ste D
Lawrence, KS 66049 USA

Publisher

Brian Osborn

Customer Service / Subscription

For USA and Canada:
Email: cs@linuxpromagazine.com
Phone: 1-866-247-2802
(Toll Free from the US and Canada)

For all other countries:
Email: subs@linux-magazine.com

www.linuxpromagazine.com – North America

www.linux-magazine.com – Worldwide

While every care has been taken in the content of the magazine, the publishers cannot be held responsible for the accuracy of the information contained within it or any consequences arising from the use of it. The use of the disc provided with the magazine or any material provided on it is at your own risk.

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.

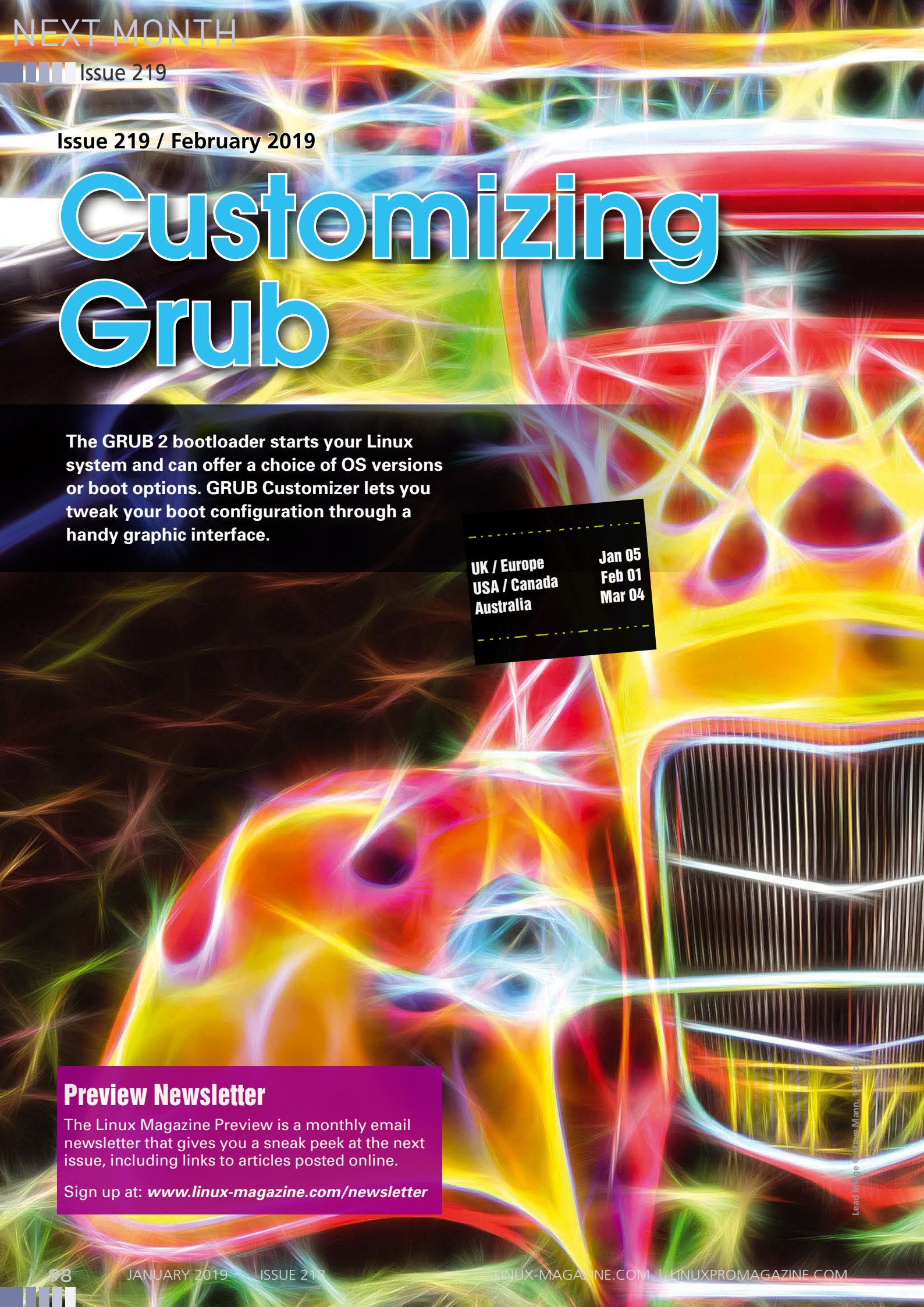
All brand or product names are trademarks of their respective owners. Contact us if we haven't credited your copyright; we will always correct any oversight.

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, 2721 W 6th St, Ste D, Lawrence, KS, 66049, USA. Periodicals Postage paid at Lawrence, KS and additional mailing offices. Ride-Along Enclosed. POSTMASTER: Please send address changes to Linux Pro Magazine, 2721 W 6th St, Ste D, Lawrence, KS 66049, USA.

Published monthly in Europe as Linux Magazine (ISSN 1471-5678) by: Sparkhaus Media GmbH, Ziebländstr. 1, 80799 Munich, Germany.



Customizing Grub

The GRUB 2 bootloader starts your Linux system and can offer a choice of OS versions or boot options. GRUB Customizer lets you tweak your boot configuration through a handy graphic interface.

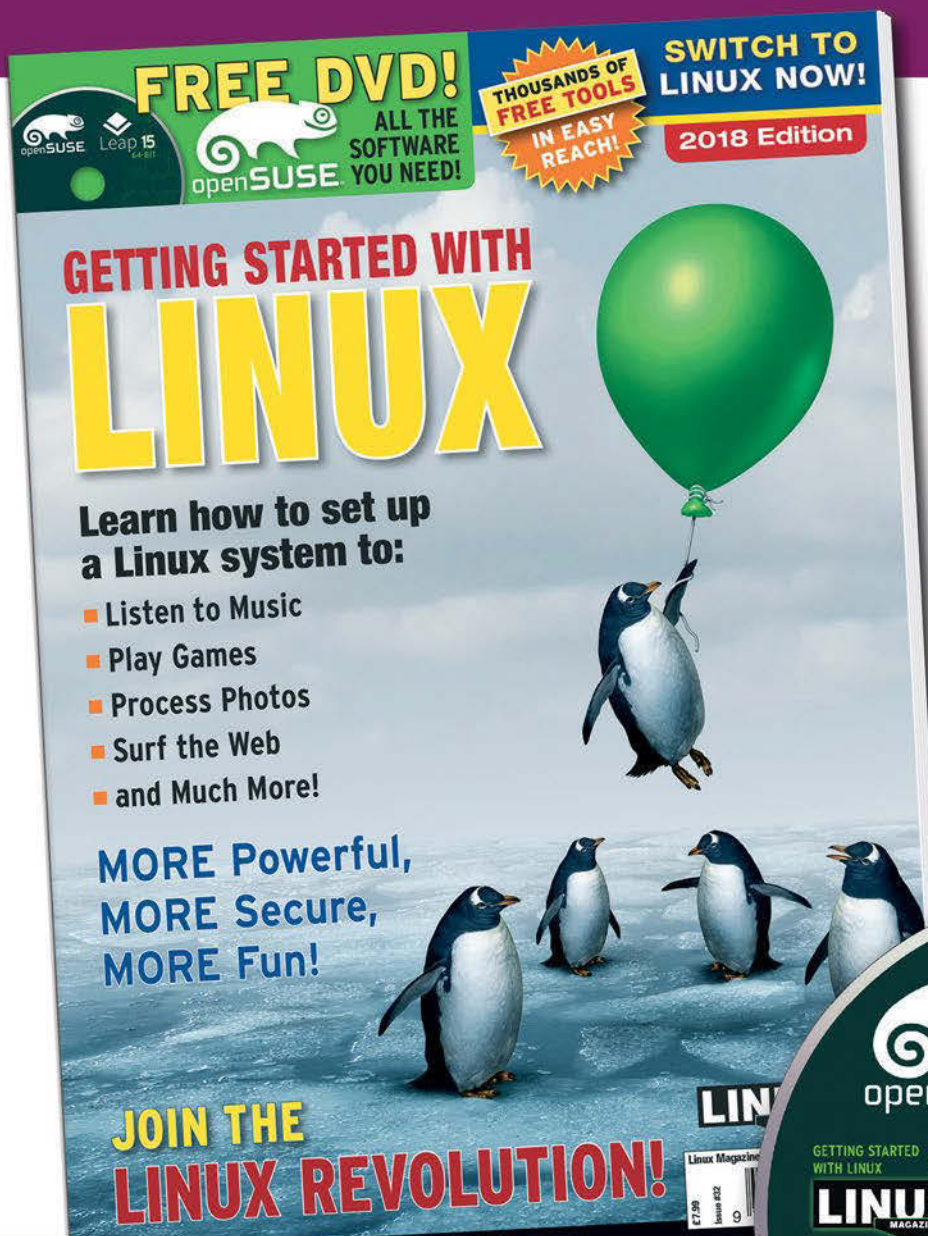
UK / Europe Jan 05
USA / Canada Feb 01
Australia Mar 04

Preview Newsletter
The Linux Magazine Preview is a monthly email newsletter that gives you a sneak peek at the next issue, including links to articles posted online.
Sign up at: www.linux-magazine.com/newsletter

Lead Image © Steve Mann, 123RF

Want your friends and colleagues to 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

TRAFFIC WITHOUT LIMITS AND FREE OF CHARGE
FOR DEDICATED ROOT AND MANAGED SERVERS



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
- ✓ Traffic unlimited
- ✓ No minimum contract
- ✓ Setup Fee £104.50

Managed Server MX92

- ✓ Intel® Core™ i7-4770 Quad-Core incl. Hyper-Threading Technology
- ✓ 32 GB DDR3 RAM
- ✓ 2 x 4 TB SATA 6 Gb/s 7200 rpm HDD Class Enterprise
- ✓ 100 GB Backup Space
- ✓ Traffic unlimited
- ✓ No minimum contract
- ✓ Setup Fee £78



monthly from £ **47.50**

monthly £ **78.00**

www.hetzner.com