FREE DVD

Double-Sided DVD
INSIDE!

WIREGUARD
Secure and simple VPN tool

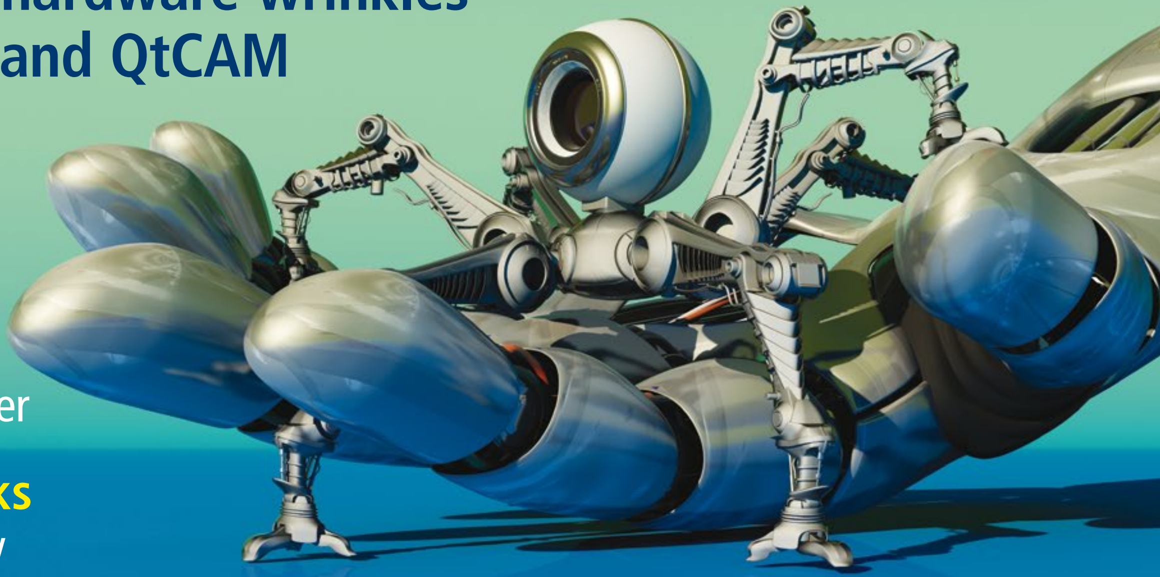
WEBCAMS AND LINUX

LINUX PRO MAGAZINE

ISSUE 237 – AUGUST 2020

WEBCAMS AND LINUX

Smoothing out hardware wrinkles with Guvcview and QtCAM



Text-to-Speech in LibreOffice

Guacamole
Remote access through a web browser

PHP Building Blocks
Assemble apps easily with PHPlattenbau

PyPortal
Extend your Pi with this portable touchscreen

Imaginary Teleprompter
Polish up your video presence

Kitchen Timer
Turn a Pi Zero into a DIY kitchen device

LINUXVOICE

- Waterfox – A faster and safer Firefox?
- maddog – COVID-19 and COBOL
- Gimp Workshop – Build a media player skin

FOSSPicks
- Audacious Music Player
- Guitar Git Client

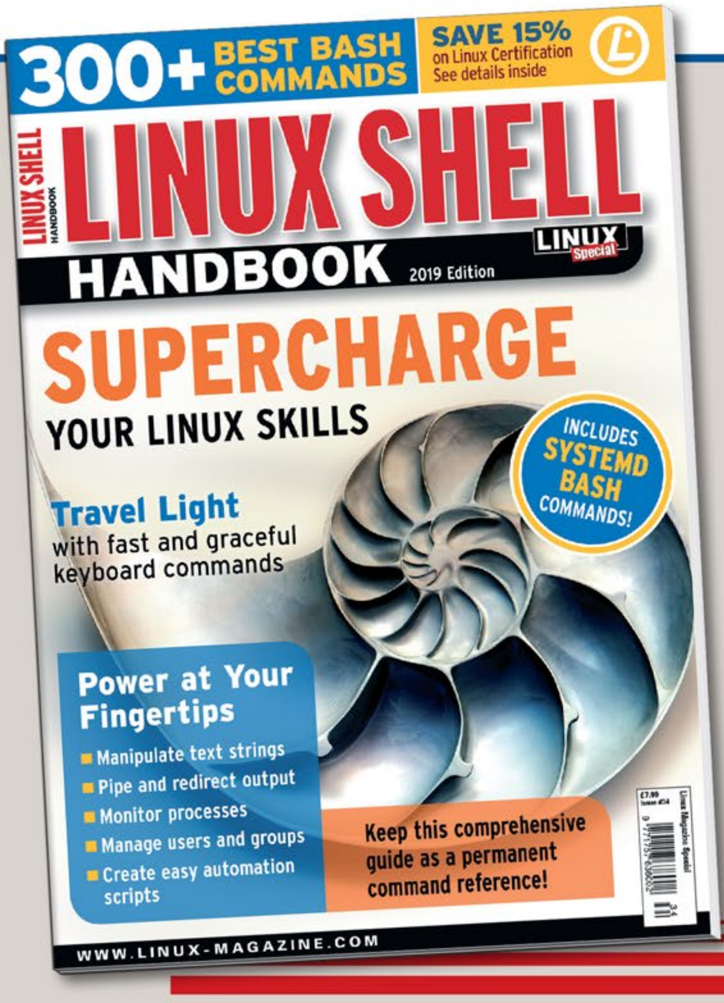
Tutorials
- MystiQ
- LÖVE Games

LINUX NEW MEDIA The Pulse of Open Source

SHOP THE SHOP
shop.linuxnewmedia.com

EXPERT TOUCH

The *Linux Shell Handbook* is available now!
This edition is packed with utilities
for configuring and troubleshooting systems.


300+ BEST BASH COMMANDS
SAVE 15% on Linux Certification See details inside
LINUX SHELL
HANDBOOK
2019 Edition
LINUX Special
SUPERCHARGE YOUR LINUX SKILLS
INCLUDES SYSTEMD BASH COMMANDS!
Travel Light with fast and graceful keyboard commands
Power at Your Fingertips
Manipulate text strings
Pipe and redirect output
Monitor processes
Manage users and groups
Create easy automation scripts
Keep this comprehensive guide as a permanent command reference!
WWW.LINUX-MAGAZINE.COM

The *Shell Handbook* provides a comprehensive look at the world inside the terminal window. You'll learn to navigate, manipulate text, work with regular expressions, and customize your Bash settings.

Here's a look at just some of what you'll see in the new Shell Handbook:

- Customizing Bash
- Pipes and Redirection
- Regular Expressions
- Text Manipulation Tools
- Systemd
- Bash Scripting
- Networking Tools
- And much more!

Make the *Shell Handbook* your permanent desktop reference on the world of the terminal window.

ORDER ONLINE:
shop.linuxnewmedia.com/specials

COPY DETAILS

Dear Reader,

The Internet Archive (IA) [1] is a nonprofit founded with the lofty goal of providing "universal access to all knowledge." Their archive of websites lets historians and journalists step back into time to view past states of the Internet. They also archive audio recordings, news programs, and digital images, making high-quality historical information available to users who might not otherwise have access.

The laudable work on the IA has received funding and accolades from leading nonprofits, such as the Andrew W. Mellon Foundation, the Democracy Fund, the National Endowment for the Humanities, and the Institute of Museum and Library Services. But the IA also sometimes receives less-glowing attention – like on June 1, for instance, when they were sued by four major book publishers in a legal challenge that could, based on the statutory limits of the alleged infractions, force the IA offline and drive it into bankruptcy [2].

A little more back story: In addition to its role as an archiver of websites, the IA is also an archiver of books. They make digital copies of print books. Sometimes they provide this service for libraries. (It is considered fair use to make an archival copy of a book you already own.) But the IA also acts as a library, loaning out electronic copies of the books in its own collection.

The original vision of the IA Open Library was to buy books and loan out an electronic version of each book one copy at a time. A reader who wanted to borrow a book that was currently on loan was put on a waiting list. As long as the IA didn't circulate more copies than it originally purchased, they saw this practice as a simple extension of the library principle.

The legality of the Open Library is still a little unclear, but many experts believe it does pass the test for fair use, so publishers learned to live with it – in part because they weren't sure they could stop it and didn't want to open the Pandora's box that could come with a full-on legal challenge.

Fast forward to last March, when the COVID-19 crisis began to close schools and libraries around the world. In an effort to fill the gap left by diminished access to published resources, the IA announced that they were "…taking the extraordinary measure to suspend wait lists on our lending collection through the duration of the US national emergency to meet the educational and inspirational needs of a global community of readers and learners." This new initiative, which they called the National Emergency Library, would not be restrained by the principle of a 1:1 ratio of licenses to simultaneous loans. In other words, they would loan multiple copies of a single purchased book.

This bold development tipped the delicate legal balance through which the Open Library operated, and the publishers reacted swiftly, stating that the IA's actions "…grossly exceed legitimate library services, do violence to the Copyright Act, and constitute willful digital piracy on an industrial scale" [3].

Many in the Free Software community have an ambivalent view of copyright law, seeing it as yet another archaic legal doctrine with no relevance to the 21st century. But regardless of what you might think about copyright as a concept, it really is a thing. Just because you might think a law *shouldn't* exist doesn't mean it *doesn't* exist. I can't tell you how many software copiers (and digital music copiers) I've talked to through the years who truly believe the edifice of copyright law is on some kind of shaky free speech ground and is about to tumble down. This won't happen. Large-scale ventures built around radical new interpretations of copyright law seriously don't work. It didn't work for Napster [4], it didn't work for Kim Dotcom [5], and it didn't work for Aereo [6]. If you feel called to distribute copyrighted material to your friends on the playground, you might be able to fly under the radar. But if you try this trick on a global scale with the blind faith that you're protected by some over-arching legal principle, you're probably going to get sued – and you might even get prosecuted.

On June 10, the IA abruptly ended their experiment with the National Emergency Library, citing the publishers' lawsuit as a reason for pulling the plug. As of this writing, it isn't clear whether the publishers will suspend their suit and seek a return to the tenuous equilibrium of the pre-pandemic status quo. The IA is apparently afraid they've poked a bear, warning that the lawsuit is not just about the National Emergency Library but "…attacks the concept of any library owning and lending digital books…" [7].

Now that controlled lending has been restored, we'll hope that the publishers will recognize the many good things the IA is doing in other areas and will seek a resolution that will allow the IA to continue with their mission of expanding and preserving the world's knowledge.

Joe

Joe Casad,
Editor in Chief

Info

[1] Internet Archive: *https://archive.org/*

[2] Lawsuit over online book lending could bankrupt Internet Archive: *https://arstechnica.com/tech-policy/2020/06/publishers-sue-internet-archive-over-massive-digital-lending-program/*

[3] Publishers' complaint: *https://www.courtlistener.com/recap/gov.uscourts.nysd.537900/gov.uscourts.nysd.537900.1.0.pdf*

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

[5] Kim Dotcom: *https://en.wikipedia.org/wiki/Kim_Dotcom*

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

[7] Internet Archive blog post: *https://blog.archive.org/2020/06/10/temporary-national-emergency-library-to-close-2-weeks-early-returning-to-traditional-controlled-digital-lending/*

LINUX MAGAZINE

WHAT'S INSIDE

This month we explore webcams, screencasting, and a cool teleprompter tool. Also inside this month's issue:

- **PHP Building Blocks** – The PHPlattenbau project offers useful components for building PHP applications (page 34).
- **Guacamole** A clientless remote access tool based on HTML5 (page 40).

Check out MakerSpace for a look at how to make your Pi Zero into a kitchen timer, and turn to LinuxVoice for a study of the handy MystiQ Audio/Video conversion tool.

SERVICE

3 Comment
6 DVD
95 Back Issues
96 Featured Events
97 Call for Papers
98 Preview

NEWS

08 News
- Linux Mint Drops Snap
- Lenovo Upping Their Linux Support
- LPI Launches FOSSlife Website
- Linux Apps on Windows
- Tuxedo Computers Joins the Ryzen Bandwagon

12 Kernel News
- When a Security Hole Is OK
- Kernel Documentation Updates
- Security Through Obscurity

REVIEWS

28 Distro Walk: Inside Linux Distros
The Linux distribution you load on your computer was developed by real people operating through community groups and other organizational structures. Read on for a glimpse into the inner workings of some leading Linux projects.

COVER STORIES

16 Imaginary Teleprompter
A teleprompter can help you give a polished look to your speeches and video presentations. Imaginary Teleprompter is a free tool that delivers professional teleprompting capabilities.

20 VokoscreenNG
The VokoscreenNG screencast tool offers many options but is still surprisingly easy to use.

24 Webcam Rescue
If your new webcam doesn't work with the default software on your Linux system, try your luck with Guvcview or QtCAM.

IN-DEPTH

30 Programming Snapshot – PDFtk Go Variant
Go is suitable for complex server programs, but it also cuts a fine figure with simple command-line tools for automating everyday life. Mike Schilli restructures the signature of a PDF manipulation app.

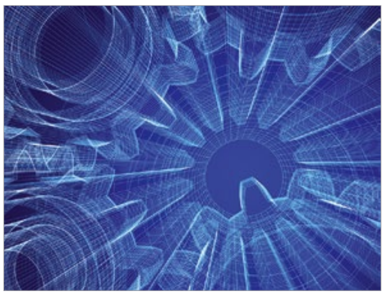
IN-DEPTH

34 PHPlattenbau
Build simple PHP applications quickly with ready-to-use blocks.

38 Charly's Column – rss2email
In order to keep up to date with security, Charly uses RSS feeds. He lets rss2email send the most important feeds directly to his mailbox to ensure that nothing is overlooked.

40 Guacamole
Use Apache Guacamole to connect to remote servers from within a web browser.

46 WireGuard
A recent addition to the Linux kernel, WireGuard lets you build a VPN tunnel that relies on encryption to reduce potential security issues.

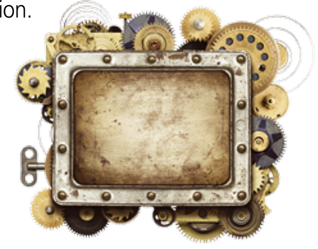


52 Text to Speech
Visually impaired users often find it difficult to work with text and tables in office suites. Pico TTS, a text-to-speech synthesizer, and the Read Text extension for LibreOffice and OpenOffice provide a solution.

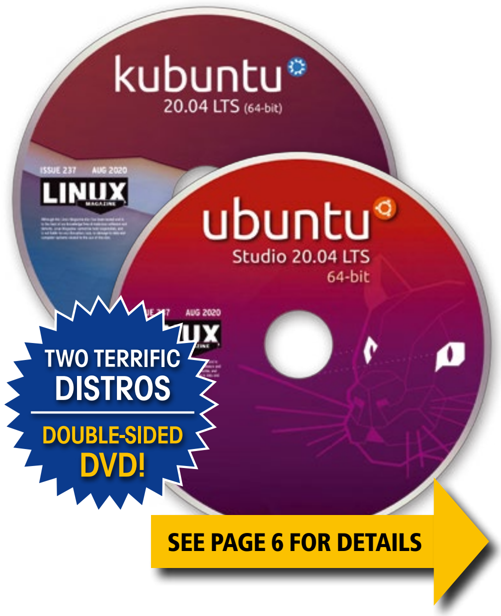
54 Command Line – Git Version Control
If you develop open source software, you need to know how to use Git. Information on Git fills books, but this basic overview will get you started with a vital open source development tool.

MAKERSPACE

58 Adafruit PyPortal
Unlike other displays for the Raspberry Pi, Adafruit's PyPortal touchscreen provides an autonomous environment, including a microprocessor, sound output, and a WiFi connection.



66 Kitchen Timer
A simple kitchen helper with two timers assists budding chefs in coping with dishes that are unlikely to be ready at the same time.



LINUXVOICE

71 Welcome
This month in Linux Voice.

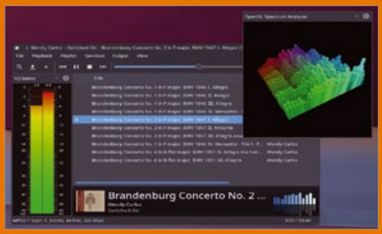
72 Doghouse – Old Systems
As unemployment claims surge, US computer systems are straining under the increased load. maddog weighs in on COVID-19 and COBOL.

74 Waterfox
This fork of the Mozilla Firefox browser is designed for greater speed and privacy, as well as compatibility with older add-ons.

76 Gimp Workshop: Media Player Skin
Forget the photos – Gimp is also a great tool for graphic design. This workshop shows how to use Gimp to create a simple visual image for a multimedia player application.

80 MystiQ
Working with multimedia files usually involves converting from one format to another. MystiQ is a fast and easy tool for file conversion.

84 FOSSPicks
This month Graham looks at Horizon EDA, ksnip, Pi-hole 5, and more!



90 Tutorial – LÖVE
To create an action-packed game with LÖVE, you'll need to overlay images to "physical" objects, detect collisions, and get input from the keyboard or mouse.

Ubuntu Studio 20.04 and Kubuntu 20.04

Two Terrific Distros on a Double-Sided DVD!

Ubuntu Studio 20.04 LTS

64-bit

Ubuntu Studio is a version of Ubuntu that is outfitted and tuned for creative artists. Onboard the default configuration, you'll find tools for photographers, graphic artists, video production specialists, and audio engineers. Although most of these applications are also available for other Linux systems, Ubuntu Studio ties them neatly into a complete package and manages configuration settings for seamless access.

For audiophiles, Ubuntu Studio boots up with the Jack low-latency audio and midi server, the Ardour digital audio workstation, the Carla virtual patchbay, and a variety of sequencers and synthesizers. Graphic artists will find the Blender 3D creation suite and the PikoPixel pixel art studio. Photographers will appreciate the Darktable virtual light table and darkroom app, and video makers will find a ready-made collection of useful tools for video editing, animation, and post-production processing.

Kubuntu 20.04 LTS "Focal Fossa"

64-bit

Kubuntu is an official Ubuntu project featuring the KDE Plasma desktop. The latest release is a Long Term Support (LTS) version. In the case of Kubuntu, LTS means the Kubuntu 20.04 release will receive support and updates for three years. The latest Kubuntu comes with the Plasma 5.18 desktop. Recent improvements include a new global edit bar that gives quick access to widgets, activities (workspaces), and configuration options. The latest release also provides better support for GTK applications, improved notification features, and enhancements to the Discover software manager.

A new Kubuntu also means an update to the popular KDE Applications suite, with more than 200 desktop applications for networking, multimedia, office productivity, and more. The KDE Applications 19.12.3 bundle rolled into Kubuntu 20.04 includes updates to familiar tools such as Kmail, KTorrent, KMyMoney, Kdenlive, the Kontact personal information manager, and other signature applications of the KDE collection.

Under the desktop, Kubuntu is very much like the other Ubuntu 20.04 flavors, with Linux kernel 5.4, as well as NetworkManager enhancements and built-in support for the WireGuard VPN manager.

Additional Resources

[1] Ubuntu Studio: *https://ubuntustudio.org/*

[2] Ubuntu Studio Community Help: *https://help.ubuntu.com/community/UbuntuStudio*

[3] Kubuntu: *https://kubuntu.org/*

[4] KDE: *https://kde.org/*

[5] Kubuntu Support: *https://kubuntu.org/*

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

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

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

Expand your Linux skills with:

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

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

Subscribe now!

shop.linuxnewmedia.com/subs

GET IT NOW!
FAST DELIVERY WITH OUR PDF EDITION

NEWS

Updates on technologies, trends, and tools

THIS MONTH'S NEWS

08 • Linux Mint Drops Snap
• Lenovo Upping Their Linux Support

09 • LPI Launches FOSSlife Website
• More Online

10 • Support for Linux Apps on Windows Is Coming
• Tuxedo Computers Joins the Ryzen Bandwagon

Linux Mint Drops Snap

In a move that surprised many within the Linux landscape, Linux Mint (one of the most popular desktop distributions) has decided to drop support for the universal snap package system (*https://snapcraft.io/*).

What are snap packages? Simply put, they are a way to combine an application and all of its dependencies into a single package. By doing this, an application can be installed on any supporting operating system, regardless of desktop or default package manager.

The idea of leaving behind snap packages began in 2019, when Clement "Clem" Lefebvre said, "When Snap was announced it was supposed to be a solution, not a problem." Clem continues, "It was supposed to make it possible to run newer apps on top of older libraries and to let third-party editors publish their software easily towards multiple distributions, just like Flatpak and AppImage."

That sentiment came to a boil recently, when Clem said, "... in the Ubuntu 20.04 package base, the Chromium package is indeed empty and acting, without your consent, as a backdoor by connecting your computer to the Ubuntu Store." In other words, Clem makes the claim that if you issue the command `sudo apt-get install chromium -y`, instead of it installing the .deb package, it instead installs the Chromium snap package. To this, Clem says, "Applications in this store cannot be patched, or pinned. You can't audit them, hold them, modify them or even point snap to a different store. You've as much empowerment with this as if you were using proprietary software, i.e. none. This is in effect similar to a commercial proprietary solution, but with two major differences: It runs as root, and it installs itself without asking you."

For users who want to continue with Mint, and would like to use snap packages, you can always install snapd after installing Linux Mint 20. But don't expect much in the way of app store integration, like that found in Ubuntu.

Original source: *https://blog.linuxmint.com/?p=3766*

Lenovo Upping Their Linux Support

PC giant Lenovo is bringing serious support to Linux ... big support. The entire line of Lenovo workstations (minus the IdeaPad) will now be fully certified to work with Linux. That's not all. Lenovo will also start selling the entire line of ThinkStation PCs and ThinkPad P series laptops with either Ubuntu LTS or Red Hat Enterprise Linux pre-installed.

Lenovo will also include full web support (*https://techtoday.lenovo.com/ww/en/workstations/linux*) and add dedicated Linux forums (*https://forums.lenovo.com/t5/Linux-Operating-Systems/ct-p/lx_en#link=%7B*) into the mix.

For many within the Linux community, this could be the biggest piece of news to develop for the open source operating system. Lenovo adding their support behind Linux not only gives consumers far more options for Linux hardware, it could easily help companies to realize the open source operating system is a viable option for the desktop. And with Lenovo also adding their drivers into the upstream kernel, all of their hardware will work with Linux out of the box. No more tweaking or compiling to get features like Wi-Fi, sound, and fingerprint readers to work.

Of this move, Rob Herman, General Manager, Executive Director Workstation & Client AI Group, said, "While many users prefer to customize their own machines – either on hardware without an OS or by wiping an existing client OS, then configuring and installing Linux – this can raise uncertainty with system stability, restricted performance, compatibility, end-user productivity and even IT support for devices." Herman added, "Now that these users are making their way out of the proverbial shadows and onto the enterprise floor, the demand is high for an out-of-the-box solution that removes the barrier for deployment of enterprise-grade hardware within a Linux software ecosystem."

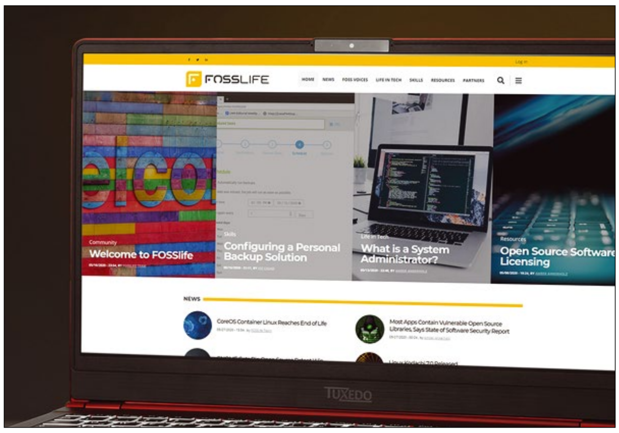
Expect Lenovo Linux preinstalled PCs and laptops to be available for order this month (June, 2020).

Original source: *https://news.lenovo.com/pressroom/press-releases/lenovo-brings-linux-certification-to-thinkpad-and-thinkstation-workstation-portfolio-easing-deployment-for-developers-data-scientists/*

LPI Launches FOSSlife Website

Linux Professional Institute (LPI) has launched FOSSlife, a website for those "who care about the FOSS community and want to follow the trends, tools, projects, programs, and people who define the FOSS experience."

The new website will offer recent news and articles on FOSS technology and advocacy (*https://www.lpi.org/articles/welcome-fosslife-new-web-magazine-born*). FOSSlife is intended to be a destination and resource for experts as well as those just starting out on their open source journey.


FOSSLIFE
Welcome to FOSSlife
Configuring a Personal Backup Solution
What is a System Administrator?
Open Source Software Licensing
NEWS
TUXEDO

"It is our mission to promote the use of free and open source by elevating the people who work with it. FOSSlife fits perfectly into this mission, as it helps us share, bundle, and disseminate knowledge about free and open source software and inspire people who are searching for their own approach in gaining this expertise," said G. Matthew Rice, Executive Director of the Linux Professional Institute.

You can check out the new site at *https://www.fosslife.org/*.

MORE ONLINE

Linux Magazine
www.linux-magazine.com

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

Bashtop, a Gorgeous Top-Like Tool
• Jeff Layton
Fairly recently, a new top-like tool has received quite a bit of attention: bashtop, like other top-like tools, shows stats for the processor, memory, disks, network, and processes. Although it might sound like any other top tool, the interface is exceptional.

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

Building a Low-Powered NAS
• Mayank Sharma
Build a network-attached storage box with Rockstor to manage your data.

Export and Analyze Azure AD Sign-In and Audit Logs
• Florian Frommherz
The relevant sign-in and audit logs in Azure Active Directory can be exported to external data sources to provide not only long-term archiving, but also the freedom to analyze the stored data.

Secure Your Data Channel with Stunnel
• Thorsten Scherf
Stunnel provides a TLS wrapper with extensive configuration options to secure your data over insecure wireless networks.

Support for Linux Apps on Windows Is Coming

At the 2020 Microsoft Build conference, it was announced that GUI applications from Linux will be available to run on WSL. This eye-opening feature comes by way of Wayland and RDP, which will draw the apps on the Windows desktop. It was also announced that access to GPUs from Linux is on the way.

As of Windows 10 20H1, insiders are able to test the feature. However, testing availability for the general public has been delayed.

Back in 2018, Whitewater Foundry created a Debian-based distribution, named WLinux, which used a Windows X server to do this very thing. Now, according to Microsoft, "support for Linux graphical user interface (GUI) apps will enable you to open a WSL instance and run a Linux GUI app directly without the need for a third-party X server."

Hayden Barnes, Canonical senior developer advocate, indicated the company had been looking to include an X Server in their Ubuntu distribution (found in the Microsoft store). Instead of putting in that work alone, Canonical opted to collaborate with Microsoft to make it happen.

With the addition of GPU support, GPU accelerated workflows (such as TensorFlow on Microk8s, running on WSL) will be unlocked.

Users should expect to still have to use the command line to make GUI apps happen within WSL. At the same time, however, Microsoft has intimated that the WSL installation process will be made simpler.

Original source: *https://blogs.windows.com/windowsdeveloper/2020/05/19/developing-for-all-1-billion-windows-10-devices-and-beyond/*

Tuxedo Computers Joins the Ryzen Bandwagon

Not one to rest on reputation, Tuxedo Computers (*https://www.tuxedocomputers.com/*) has upped the ante for their Linux pre-installed options. This time around, the Linux-only computer manufacturer has released the first-ever Ryzen-powered Linux laptop. The Tuxedo Book BA15 has one option for CPU – the AMD Ryzen 5 3500. As for GPU, the BA15 ships with the AMD Radeon Vega 8.

This could be considered significant, given that only days ago Linus Torvalds (the creator of Linux) announced he'd moved his main machine away from an Intel CPU to an AMD Ryzen. Torvalds claims the AMD Threadripper 3970x has his test builds of the kernel running three times faster than they were with the Intel chip.

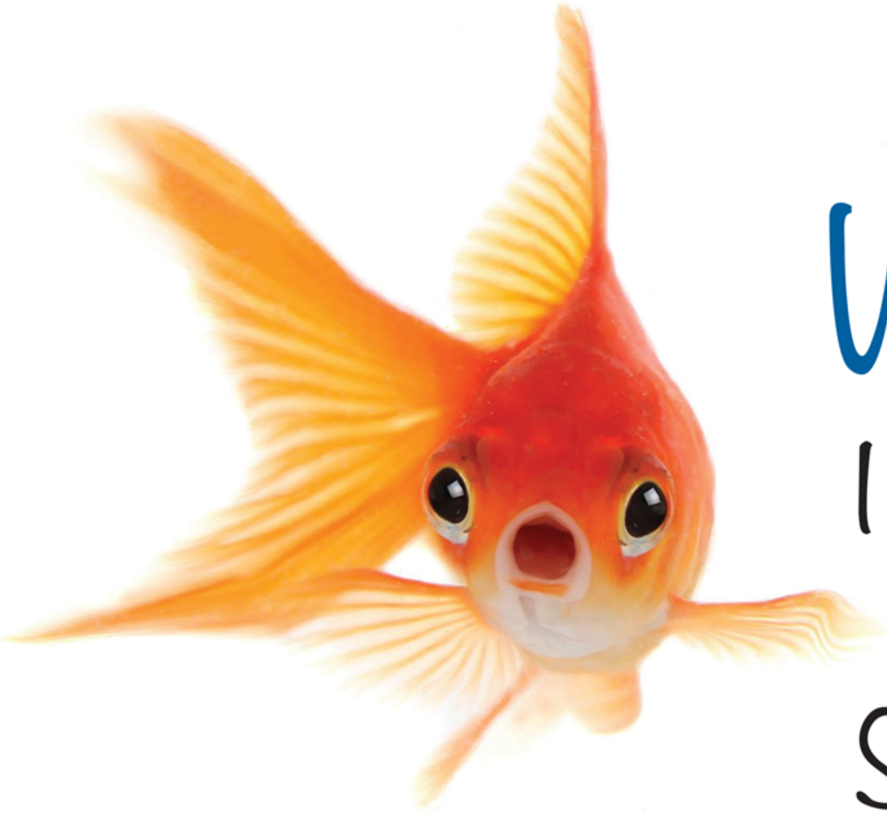
Make no mistake, the Ryzen 5 3500 is no Threadripper. Instead of the massive 32 cores found in the 3970x, the 3500 has a meager 6 cores. However, the 3500 is no slouch and should be a perfect match for Linux on a mobile device. Add into the mix a massive 91Wh battery and you could get up to 13 hours of office work (including web, email, and word processing), with 10 hours of full HD 1080p video streaming at 50% display brightness.

The Tuxedo Book also features a 15" display, full-sized backlit keyboard, Bluetooth 5.1, wireless ac/a/b/g/n/ax compatibility, 1 USB C port, 2 3.2 Gen1 USB A port, 1 USB 2.0 A port, 1 Gigabit LAN RJ45 port, a 9-in-1 card reader, and up to 32GB of RAM. The base price of the BA15 is 859 EUR ($936.00 USD). Available for purchase here: *https://www.tuxedocomputers.com/en/Linux-Hardware/Linux-Notebooks/15-16-inch/TUXEDO-Book-BA15-Gen10.tuxedo*.

Get the latest news in your inbox every two weeks

Subscribe FREE to Linux Update

bit.ly/Linux-Update



Sign up for a digital subscription to improve our admin skills wth practical articles on network security, cloud computing, DevOps, HPC, storage and more!

Subscribe to the PDF edition: **https://bit.ly/digital-ADMIN**

Now available on ZINIO: bit.ly/ADMIN-ZINIO

Zack's Kernel News

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

By Zack Brown

When a Security Hole Is OK

Eric W. Biederman recently posted a patch to replace a 32-bit counter with a 64-bit counter. This would fix the problem that, as he put it, "With care an attacker can cause exec_id wrap and send arbitrary signals to a newly exec'd parent."

He added that he had tested this hole and found that he could wrap the 32-bit `exec_id` and exploit the problem in two weeks. Faster systems, of course, could do it more quickly.

However, Eric did acknowledge that on 32-bit CPUs, "reading self_exec_id is no longer atomic and can take two read instructions." This meant that on 32-bit systems there would be a microscopic window of time when the actual `self_exec_id` value would not match the value being read by the code. During that time, he said, this security hole remained exploitable.

Linus Torvalds acknowledged the patch, saying it didn't seem urgent, and asked Eric to put it in his own source tree, where it would percolate up to Linus the next time he took a merge from that tree.

Jann Horn was concerned. Jann wrote some test code that reduced Eric's 14-day rollover to 14 hours.

Jann said that while 64-bit CPUs would probably be fine, 32-bit CPUs would be vulnerable to "store tearing." Store tearing is the name for what Eric described – when it takes two instructions to write a piece of data, and those two instructions can be "torn" by malicious code to take advantage of the fact that the actual data is different from what we think it is in that brief instant. The reciprocal term "load tearing" refers to when it takes two instructions to read a piece of data.

But Linus said he didn't care.

He said that in order to exploit that vulnerability, "first you'd have to work however many weeks to do 4 billion execve() calls, and then you need to hit basically a single-instruction race to take advantage of it. Good luck with that. If you have that kind of God-like capability, whoever you're attacking stands no chance in the first place."

Jann agreed the risk was low, but said the code was simply technically wrong and amounted to having an "intentional race" condition in the kernel. This could theoretically break tools that tested kernel security. It could also lead developers to unwittingly copy that same broken code for use elsewhere. Jann suggested at least including a code comment to let people know this wasn't how to do things.

Even better, Jann said, would be that, "Since the read is already protected by the tasklist_lock, an alternative might be to let the execve path also take that lock to protect the sequence number update, given that execve is not a particularly hot path."

Linus agreed that a code comment would be fine. However, `tasklist_lock` was very time consuming and already highly utilized in the kernel and wasn't worth it for a security hole that could essentially never be exploited.

Eric asked Linus which code paths used `tasklist_lock` so much. He offered to go through those code paths and clear out some of the locks if Linus thought it was a problem.

Linus explained:

"It's generally not bad enough to show up on single-socket machines.

*"But the problem with tasklist_lock is that it's one of our remaining completely global locks. So it scales like sh*t in some circumstances.*

"On single-socket machines, most of the truly nasty hot paths aren't a huge problem, because they tend to be mostly readers. So you get the cacheline bounce, but you don't (usually) get much busy looping. The cacheline bounce is 'almost free' on a single socket.

"But because it's one of those completely global locks, on big multi-socket machines people have reported it as a problem forever. Even just readers can cause problems (because of the cacheline bouncing even when you just do

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**.

the reader increment), but you also end up having more issues with writers scaling badly.

"Don't get me wrong – you can get bad scaling on other locks too, even when they aren't really global – we had that with just the reference counter increment for the user signal accounting, after all. Neither of the reference counts were actually global, but they were just effectively single counters under that particular load (ie the count was per-user, but the load ran as a single user).

"The reason tasklist_lock probably doesn't come up very much is that it's _always_ been expensive. It has also caused some fundamental issues (I think it's the main reason we have that rule that reader-writer locks are unfair to readers, because we have readers from interrupt context too, but can't afford to make normal readers disable interrupts).

"A lot of the tasklist lock readers end up looping quite a bit inside the lock (looping over threads etc), which is why it can then be a big deal when the rare reader shows up.

"We've improved a _lot_ of those loops. That has definitely helped for the common cases. But we've never been able to really fix the lock itself."

The conversation ended there.

It seems as though Jann has made a good case that the opportunity to encounter the race condition can be accomplished a lot quicker than Eric first thought. Though as Linus pointed out, an attacker would need to wrap the 32-bit `exec_id` variable over and over again in order to hit the actual race condition.

I often talk about how in Linux development security trumps all other considerations. And this is true, but it doesn't mean that all security holes get plugged. In a situation like this, for example, Linus made the judgment that "If you have that kind of God-like capability, whoever you're attacking stands no chance in the first place." It sounds a little flippant, but it's an actual principle of kernel development.

The same principle accounts for why Linus has sometimes refused to add security patches for cases when an attacker has physical access to a machine.

In those cases, Linus's assumption is, if someone has physical access to your machine, then that's the ball game. You've been owned. And trying to throw tiny obstacles into the path of that owning is simply pointless.

This is one reason why there is a contingent of hackers who feel that Linus is bad on security. That contingent believes it's important to shrink all possible attack surfaces as small as possible, whereas Linus doesn't care about security issues that are only theoretical, but impossible in practice, or that he considers irrelevant, because the conditions creating the vulnerability would also create other much larger vulnerabilities.

In general, if a security hole can't be exploited, Linus doesn't care about it. And if a security hole can only exist in the context of a much larger security hole, Linus won't care about the smaller hole if the larger hole remains unpatched.

Kernel Documentation Updates

There was recently some debate over documentation file formats. John Mathew posted some documentation for the scheduler, using reStructuredText format (RST) and DOT (a graph description language).

There will always and forever be holy wars fought over file formats. This time, Peter Zijlstra remarked, "I despise RST; it's an unreadable mess." As an example, he pointed to some DOT code describing a graph and said "it pretty much mandates you view this with something other than a text editor."

Daniel Bristot de Oliveira, in defense of DOT, replied, "The good thing about the dot format is that we can convert it to many other formats, including text." And to prove it, he ran some of the documentation through the "graph-easy" program and produced a lovely ASCII art depiction of what was being described.

Peter replied, "Oh, I know and love dot files; I generate them occasionally. But they stink as end-result, which is what it is here. If you can't read a document (or worse comment) in a code editor, it's broken (and yes, I know some subsystems have a different opinion here)."

And Valentin Schneider remarked, "Agreed. Feel free to use dot to draw stuff, but please include the textual version in the doc. If you really insist on having a fancier image in the web output, have a look at things like ditaa (or whatever equivalent is supported in rst)." The ditaa program is used to convert ASCII art diagrams into graphical images.

Jonathan Corbet also suggested, "just put the ascii art into the doc as a literal block. I don't see any real reason to embed Dot stuff unless there's really no alternative."

Daniel agreed with Jonathan, saying that it might make the most sense to keep the DOT file separate and use it to generate the ASCII art for the documentation – sort of a source file generating another source file.

Meanwhile Matthew Wilcox, coming from an entirely different file format religion, mentioned to John, "please format your line lengths to more like 75 characters; don't go all the way to 80. Just use 'fmt'; its defaults work fine." But there was no discussion on that point.

Documentation is an eternal struggle, and it's a miracle any two developers ever agree on anything about it. You've got your RST people, your wiki people, your plain text people, your man page people, your code comment people, and the list goes on. And since finding a developer who wants to write documentation is sort of like finding a two-headed donkey, it's often best to just let them do what they want and not ask too many questions. But that won't stop all the other two-headed donkeys from griping about it.

Security Through Obscurity

It can be very tempting to try to secure something by making it very difficult to decipher. A lot of websites do this with JavaScript. They need the code to run on the user's system so they have to transmit it as code, but they also don't want the user to figure out what it's doing, so they run it through a "compiler" to turn it into mush before they send it.

It doesn't really work. Someone always cracks the code, and that's the end of the security. It's pretty funny to realize the number of JavaScript "compilers" and "decompilers" available in the world.

In February, Ilya Dryomov had a similar objection to a security feature in the Linux kernel. He noticed that `NULL` and `IS_ERR()` error pointers were obfuscated and remarked, "it probably wouldn't take long for an attacker to find the hash that corresponds to 0. Although harder, the same goes for most common error values, such as -1, -2, -11, -14, etc."

Rather than increasing security, he added, "Obfuscating them just makes debugging based on existing pr_debug and friends excruciating."

He posted a patch to remove the obfuscation.

The issue is made more daunting by the fact that error codes are not universal. In a later email, Ilya pointed out that the `EAGAIN` code was different on the Alpha architecture than on Intel.

However, the patches fell right down on the floor and lay there for a couple of months, until Ilya brought them up again to ask if they would be accepted or not. He pointed out that although there had been some discussion elsewhere on the mailing list, the fix itself had come through the gauntlet OK. And Ilya added, "If you want to restructure the test suite before adding any new test cases, v1 doesn't have them. Other than the test cases, the only difference between v1 and v2 is added reviews and acks."

Andy Shevchenko replied, saying that Petr Mladek, who would ordinarily be the one to accept such patches, "has some obstacles that prevent him to pay attention on this and actually do anything right now for Linux kernel. If Rasmus, Sergey, you and maybe others will got consensus here, I think Andrew can take it thru his tree."

Ilya remarked that the other two maintainers, Sergey Senozhatsky and Steven Rostedt, had already approved the patches. Ilya reminded Andy that the delay therefore was because Petr had wanted to restructure the test suite first. Of course if he wasn't available for kernel work....

Andy suggested generating the patches again, based on the latest kernel release, and submit them for the next upcoming release. Then everyone would have another chance to say yay or nay.

At this point Linus Torvalds replied, "For something simple and straightforward like the obfuscation patches, I can take them directly if people are unavailable. Just make sure to re-post the patches, and make it clear that it's to me (because usually if I'm cc'd I end up just reading and then ignoring things that I expect to go through a maintainer)."

So that was that.

The interesting thing about all this to me is the way it reveals Linus's policy towards security patches. In this case, obfuscating the error codes did provide a benefit, in that they forced an attacker to do work in order to figure them out. And to a lot of security hackers, any obstacle they can put in front of a bad actor is a good obstacle. But Linus seems to disagree. He seems to believe that obstacles are only good if they actually prevent the bad actor from penetrating system security; anything else only tends to make the code less maintainable, more bug-prone, slower, and uglier. He wants the kernel source code to be sleek and beautiful, without any extras that are there for theoretical purposes only. ■■■

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**

LINUX UPDATE

EXPLORING THE WORLD OF LINUX

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

FEATURED ARTICLES

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

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

openSUSE OBS Can Now Build Windows WSL Images
openSUSE enables developers to build their own WSL distributions. (more)

Scribus
Scribus is a desktop publishing tool for Linux that expands your options when designing simple flyers and brochures, giving you plenty of room to be creative. (more)

Tutorial – Fediverse
If you're looking for social media options where the user has more control, you'll find a range of options to explore in the Fediverse, including the popular Mastodon. (more)

FURTHER READING

- GnuCash: Bookkeeping the FOSS Way
- Red Hat Announces CentOS Stream
- Richard Stallman Resigns from Free Software Foundation
- Jump Box Security
- SystemRescueCd

Get hyped.
The International Conference for High Performance Computing, Networking, Storage, and Analysis
Nov 17–22 | Denver, CO
REGISTER →
SC19 | hpc is now.

MOST READ

chkrootkit
Linux can be infected by rootkit malware that is hidden and hard to detect. The chkrootkit program can help find rootkit infections. (more)

Linux Magazine Trial

bit.ly/Linux-Update

Ft Photography, Fotolia

Smooth out your words with Imaginary Teleprompter

Eloquence

A teleprompter can help you give a polished look to your speeches and video presentations. Imaginary Teleprompter is a free tool that delivers professional teleprompting capabilities. *By Erik Bärwaldt*

Teleprompters have become an important tool in television and politics. The goal of a teleprompter is to display text to the speaker in a way that gives the viewer the impression the speaker is improvising. The recent rise of platforms such as YouTube or Dailymotion has democratized video production, which means that more and more private video bloggers have an occasional need for teleprompters to deliver essays, speeches, and other presentations.

You don't need expensive equipment to integrate a teleprompter with your video creations. Armed with just a Raspberry Pi and the free Imaginary Teleprompter [1] software, you will have the same capabilities as speakers at a political event or on TV.

For Everyone

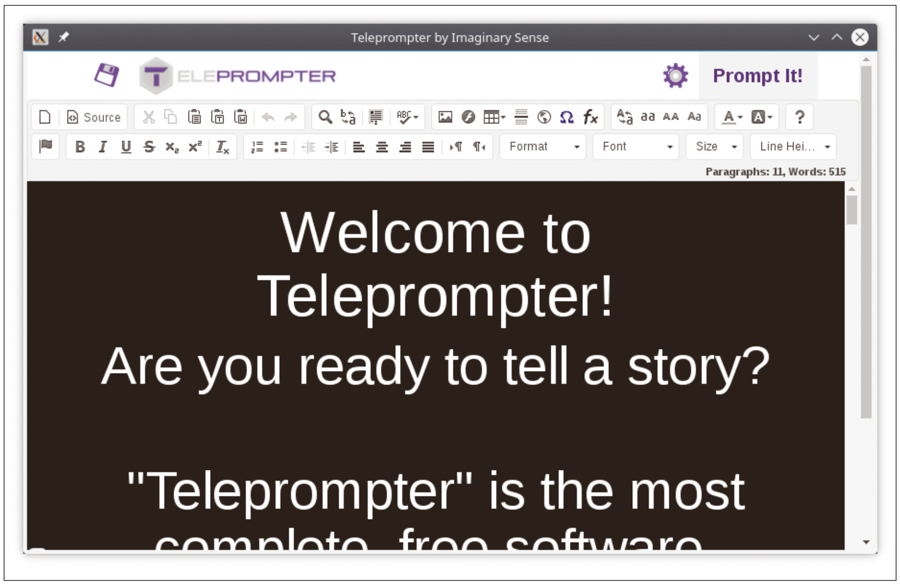
Imaginary Teleprompter is available at the project website for all common platforms. Many distributions also offer their own binary packages. Since the RaspPi does not appear explicitly in this list, the recommendation is to choose the appropriate AppImage from the *Other Linux Distros* section.

The package listed there (*ARM 7L*) is suitable for both 32- and 64-bit systems and is about 36MB in size. Type the command from the first part of Listing 1 to grant the software the required execute permissions. Then move the package to a directory of your choice and call it with the command from the second line of Listing 1.

After a prompt asking whether the package should be integrated into the menu structure of your desktop, the program window opens (Figure 1). Since Imaginary Teleprompter requires considerable resources, the program takes several seconds to start, especially on older RaspPi models. At first glance, the editor window looks like a conventional word processor. At the top, you will find two horizontal tool-

Listing 1: Installing

```
$ chmod +x imaginary-teleprompter-2.3.4-armv7l.AppImage
$ ./imaginary-teleprompter-2.3.4-armv7l.AppImage
```

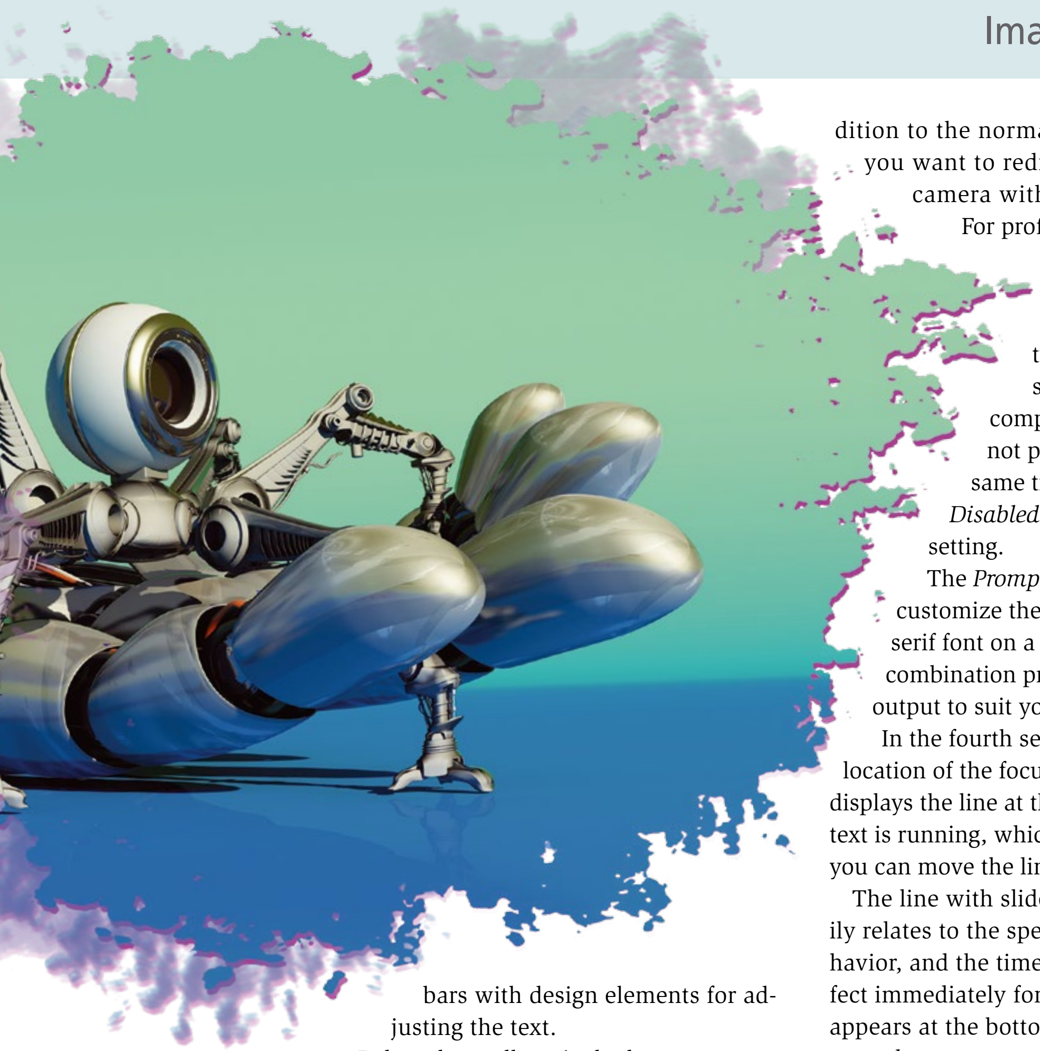


Paragraphs: 11, Words: 515

Welcome to Teleprompter!
Are you ready to tell a story?

"Teleprompter" is the most complete, free software

Figure 1: The Imaginary Teleprompter editor window looks like the interface of a standard word processor.



bars with design elements for adjusting the text.

Below the toolbars is the large text area that contains the text. And below the text is a status bar, which does not yet provide any information when the software is called up. The integrated text editor is based on the web-based CKEditor [2], which, like Imaginary Teleprompter, is available under a free license.

Customize

Click on the gear symbol in the top right-hand corner of the program window to call up the software's setup panel. You will not find a conventional configuration menu but only eight options relating to the text display (Figure 2).

In the *In-frame prompter* selection menu, you can decide how Imaginary Teleprompter should display the text. In addition to the normal output, it can also mirror the output if you want to redirect the screen output to a recording camera with a mirroring system.

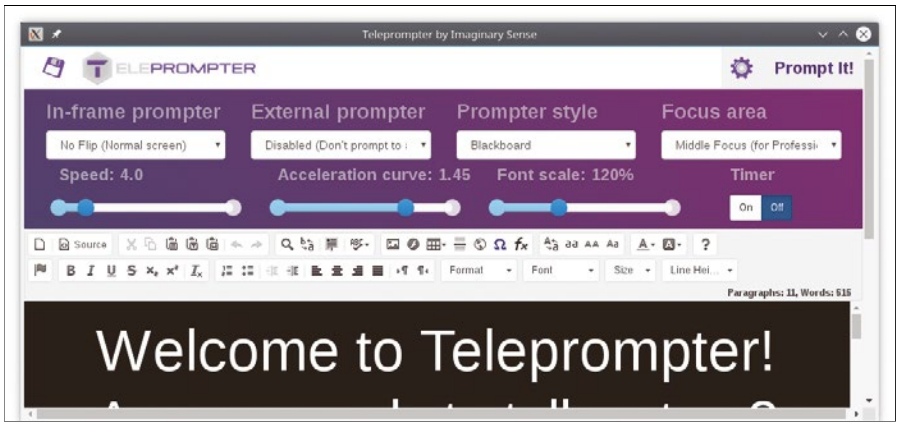
For professional recordings, there is the *External prompter* option to the right. The *External prompter* configuration options are intended for a second teleprompter, of the type used for speeches and lectures. The second teleprompter can be controlled completely independently of the first one. It is not possible to switch off both prompters at the same time. If you accidentally set both dialogs to *Disabled*, the program tells you this is an incorrect setting.

The *Prompter style* setting option lets you visually customize the prompter. By default, you will see a light serif font on a dark background. Several color and font combination presets in the program let you customize the output to suit your preferences.

In the fourth selection field, *Focus area*, you determine the location of the focus line. By default, Imaginary Teleprompter displays the line at the center of the screen brightest when the text is running, which helps the speaker orient. Alternatively, you can move the line within the playback window.

The line with sliders below the four selection fields primarily relates to the speed of the text, the font size, the fading behavior, and the timer. Modifications to these options take effect immediately for the currently loaded text; the timer only appears at the bottom of the window when prompt mode is started.

After completing the settings, click the gear symbol again to leave the options area.



Teleprompter by Imaginary Sense

In-frame prompter: No Flip (Normal screen)
External prompter: Disabled (Don't prompt to :
Prompter style: Blackboard
Focus area: Middle Focus (for Professi
Speed: 4.0
Acceleration curve: 1.45
Font scale: 120%
Timer: On Off

Paragraphs: 11, Words: 515

Welcome to Teleprompter!

Figure 2: The Prompter module configuration dialog only has the bare essentials, which keeps the interface tidy and simple.

Entering Text

You enter the text displayed later on in the teleprompter as in a conventional word processor. To do this, first click on *New Page* on the left in the button bar; the sample text then disappears. Now enter the text for the teleprompter to display as you would in any graphical word processor and format it as required.

You also have the option of importing text you have already written in LibreOffice or OpenOffice format. To import text, click on the floppy disc icon above the button bar. This button opens a vertical option bar on the left where you will find the *Import Files* and *Add Files* functions.

To import text, open the *Import Files* dialog and select the desired document in the file manager. Clicking on *Add Files* brings up a small dialog that prompts you to enter a file name. The program then uses this name for the new file that you edit in the editor area.

Editing Functions

Imaginary Teleprompter offers numerous editing options for the text. For example, you can change the type, size, and attributes of the font, as well as the text align-

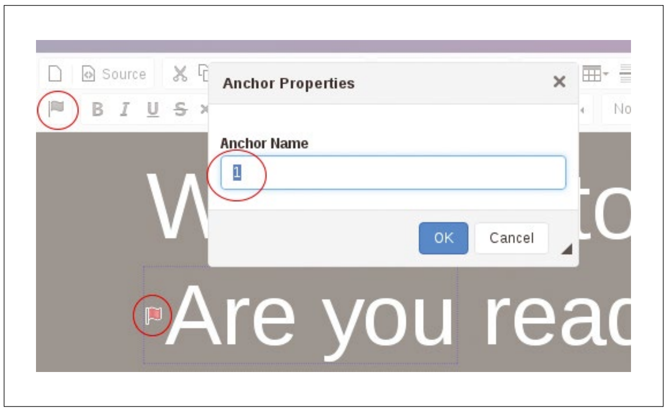


Figure 3: Anchors stored in the text let you jump to certain text passages in teleprompter mode by pressing a key.

ment if necessary. These adjustments help you visually highlight special features, such as the emphasis of individual terms.

The *Search and Replace Terms* dialog lets you edit selected text sequences or individual terms. Other editing functions include inserting and managing tables, graphics, or formulas. The editor also includes a language-specific spell checker that lets you choose a dictionary.

Even when using all the functions, the training curve is quite flat, since symbols and keyboard shortcuts are oriented on the familiar features of conventional graphic word processors. However, the editor does not support the usual menu structures and menu bars that you will find in graphic editors.

Anchor

An anchor marks a text passage or assigns it to a key. Pressing the predefined key jumps to the anchored text passages in the prompter window during text playback (Figure 3).

To add an anchor to a term, click on the small flag symbol in the lower-left button bar after highlighting the text passage. In the window that opens, enter the key you want to press to jump to this text passage as the *Anchor Name*. Pressing on *OK* saves the mapping.

Playback

After you have entered and saved the text, start the teleprompter by clicking *Prompt It!* in the top-right corner of the editor window. The software then switches to a full screen mode and all the controls disappear. If you are using the preset options, the scrolling text will start scrolling through the image from bottom to top at a steady speed.

The presentation brightens the image background from the bottom to the middle line by line in three steps and darkens it again from the middle upwards. The text therefore appears lighter towards the middle and then darkens again until it disappears at the top edge of the image.

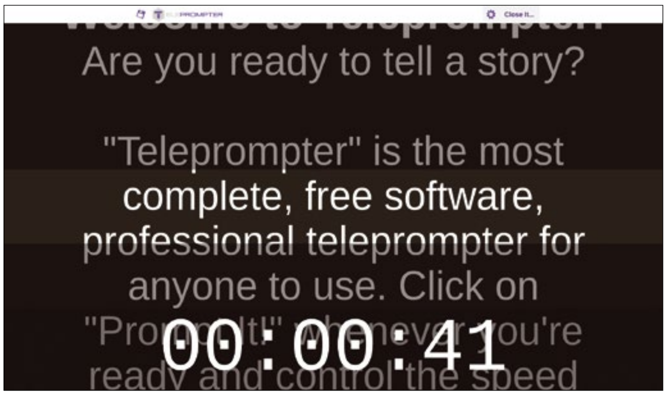
When reading, you need to concentrate on the middle, and thus brightest, line of the text display. This ensures that the text is read at a constant speed; you can compensate for slight deviations by reading the following line. If the timer is activated, it will count along at the bottom edge of the screen (Figure 4).

Several keyboard shortcuts, which are described in detail in the program documentation, are available for control functions during text playback. The shortcuts include, for example, speeding up, slowing down, and pausing the text flow, changing the font size, or even functions for controlling the timer.

After the text has been scrolled through, or if you need to stop to edit a passage, you can disable the prompter by clicking on *Close It…* at the top right of the window. Alternatively, you can close the window by pressing F8 or Esc.

Conclusions

Imaginary Teleprompter and your RaspPi save you the cost of a professional teleprompter. Even on the less powerful Model 3, the software does its job quickly and without serious delays, although starting the app images can take a good thirty seconds or even a minute on older Rasp Pi models. Imaginary Teleprompter's range of functions, and its catchy concept, ensure that you will get very usable results without too much training. ■■■


Are you ready to tell a story?

"Teleprompter" is the most complete, free software, professional teleprompter for anyone to use. Click on "Prompt It!" whenever you're ready and control the speed

00:00:41

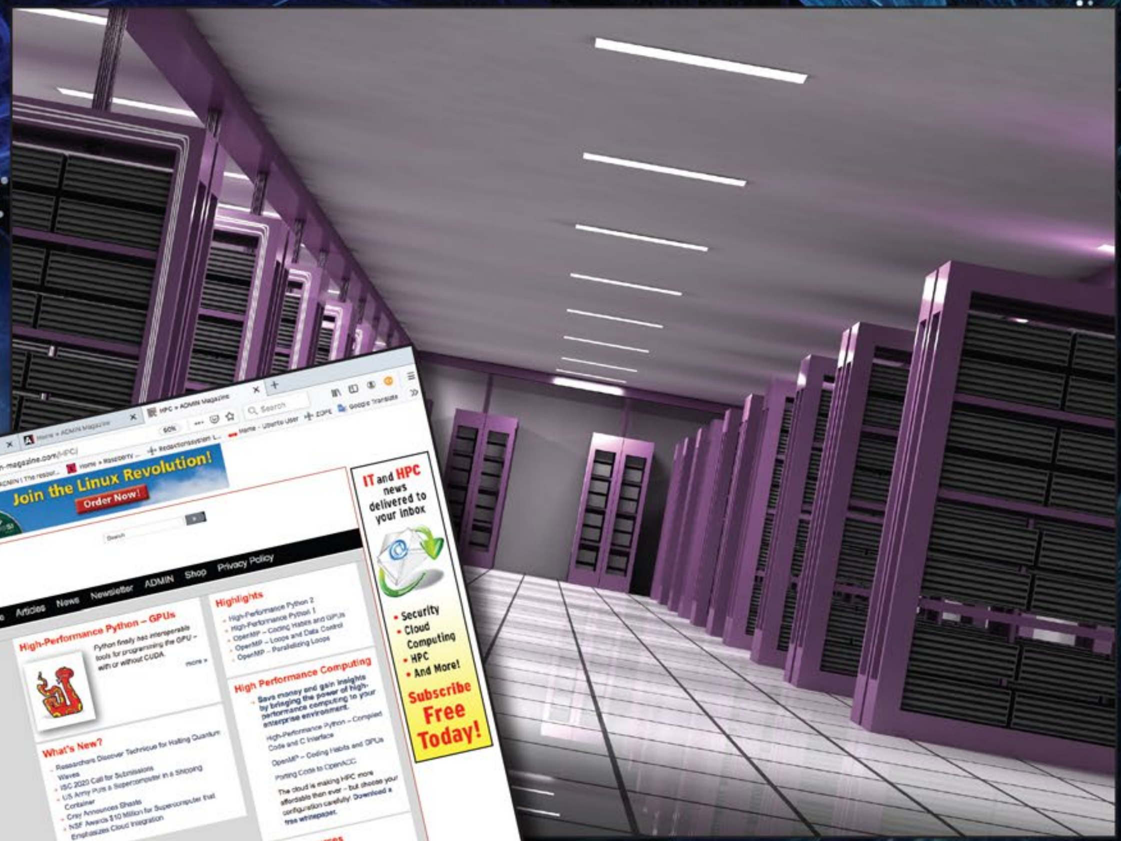
Figure 4: In teleprompter mode, Imaginary Teleprompter only displays the running text, plus a timer if enabled.

Info

[1] Imaginary Teleprompter: *https://imaginary.tech/teleprompter/*

[2] CKEditor: *https://ckeditor.com*

A Webzine for High-Performance Computing Specialists



Join the Linux Revolution! Order Now!

ADMIN HPC

Home Articles News Newsletter ADMIN Shop Privacy Policy

ARTICLES
NEWS
VENDORS
WHITEPAPERS
WRITE FOR US
ABOUT US

High-Performance Python – GPUs

Python finally has interoperable tools for programming the GPU – with or without CUDA.

more »

What's New?

- Researchers Discover Technique for Halting Quantum Waves
- ISC 2020 Call for Submissions
- US Army Puts a Supercomputer in a Shipping Container
- Cray Announces Shasta
- NSF Awards $10 Million for Supercomputer that Emphasizes Cloud Integration

Highlights

- High-Performance Python 2
- High-Performance Python 1
- OpenMP – Coding Habits and GPUs
- OpenMP – Loops and Data Control
- OpenMP – Parallelizing Loops

High Performance Computing

- **Save money and gain insights by bringing the power of high-performance computing to your enterprise environment.**

High-Performance Python – Compiled Code and C Interface

OpenMP – Coding Habits and GPUs

Porting Code to OpenACC

The cloud is making HPC more affordable than ever – but choose your configuration carefully! **Download a free whitepaper.**

HPC Resources

- OpenACC – Porting Code
- OpenACC – Data Management
- OpenACC – Parallelizing Loops
- GUI or Text-Based Interface?
- Resource Management with Slurm

IT and HPC news delivered to your inbox

- Security
- Cloud Computing
- HPC
- And More!

Subscribe Free Today!

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 Network & Security

admin-magazine.com/HPC

Record screencasts with VokoscreenNG

On Air

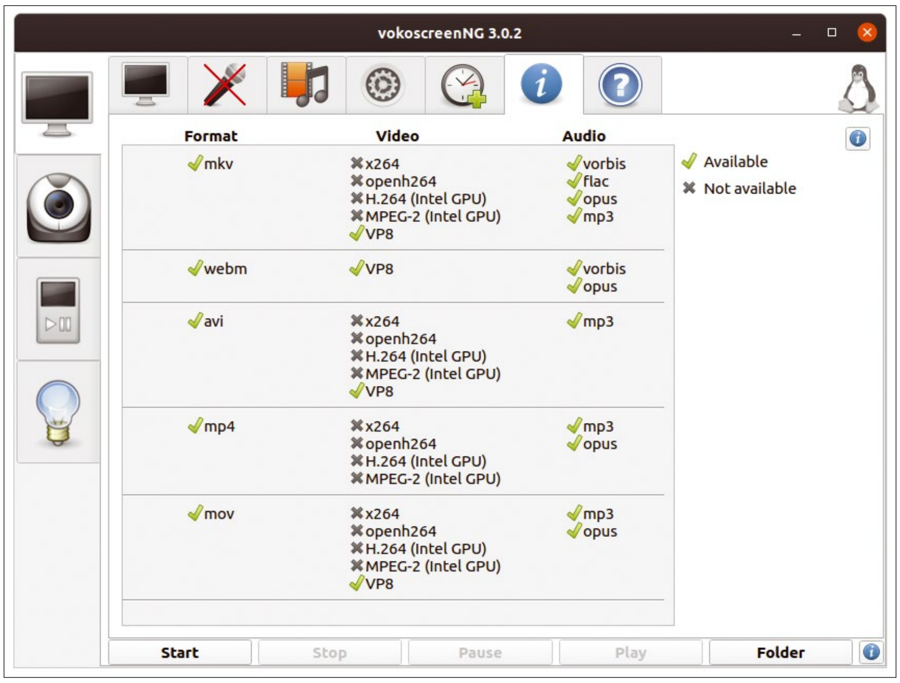
The VokoscreenNG screencast tool offers many options but is still surprisingly easy to use. *By Ferdinand Thommes*

A screencast is a screenshot with moving pictures. Like a screenshot, a screencast opens up the possibility of explaining complex processes in a way that is more effective than words. Screencasts play an important role in explaining how to use programs in both professional and private contexts.

Several screenshot applications inhabit the Linux space, including RecordMyDesktop, SimpleScreenRecorder, and the not-necessarily-intuitive OBS Studio. One screencast tool that was popular in the past and is now making a comeback is Vokoscreen. Volker Kohaupt has been developing the Vokoscreen screencast app since 2013. In 2019, development slowly stopped; after the 2.5.8 release there were no further updates. But in 2020, VokoscreenNG 3.0 was released [1]. The "NG" stands for *new generation*, and the latest version of Vokoscreen does indeed include some innovations that should make it popular with a new generation of Linux users.

The developer completely rewrote the tool and gave it a modern interface based on the Qt framework. Instead of Ffmpeg, the Gstreamer [2] multimedia framework, which already supports use with Wayland, now runs in the background. This, in turn, makes VokoscreenNG fit for cooperation with the Pipewire [3] audio-video framework, which could eventually replace Gstreamer, Pulseaudio, and Jack.

VokoscreenNG is well suited for recording on-screen processes for demonstration purposes. It can also record audio and display webcam output in a separate window. The application is also useful for recording video conferences. The resulting record-



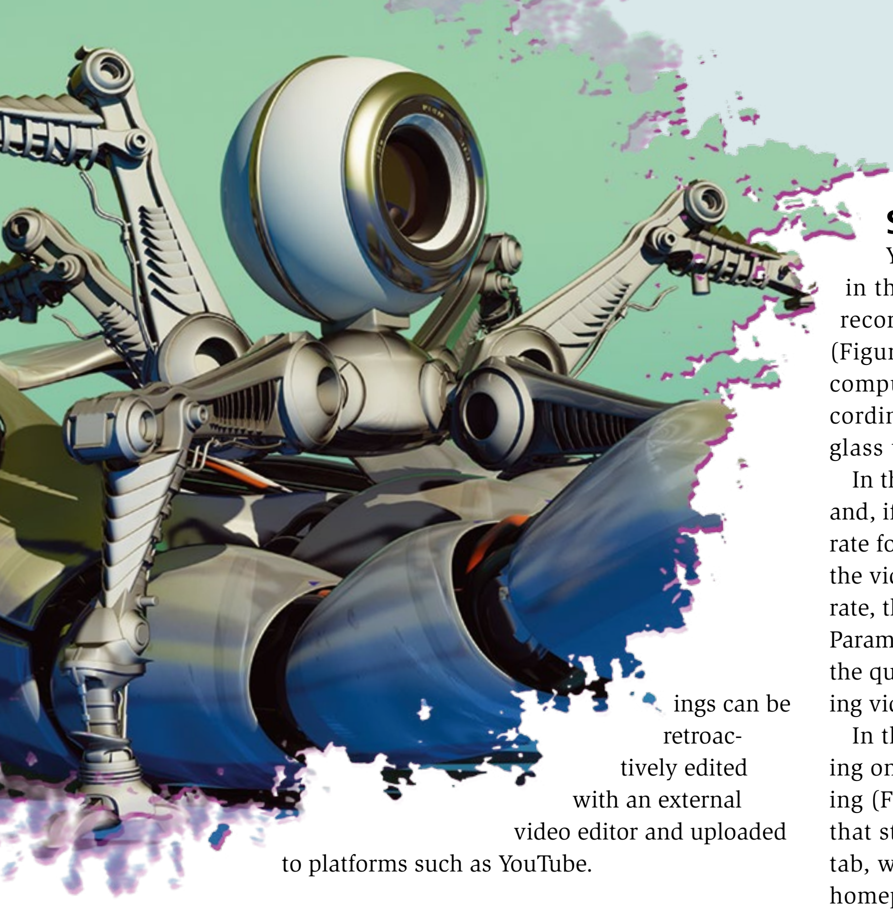
Format	Video	Audio
✓mkv	✖x264 ✖openh264 ✖H.264 (Intel GPU) ✖MPEG-2 (Intel GPU) ✓VP8	✓vorbis ✓flac ✓opus ✓mp3
✓webm	✓VP8	✓vorbis ✓opus
✓avi	✖x264 ✖openh264 ✖H.264 (Intel GPU) ✖MPEG-2 (Intel GPU) ✓VP8	✓mp3
✓mp4	✖x264 ✖openh264 ✖H.264 (Intel GPU) ✖MPEG-2 (Intel GPU)	✓mp3 ✓opus
✓mov	✖x264 ✖openh264 ✖H.264 (Intel GPU) ✖MPEG-2 (Intel GPU) ✓VP8	✓mp3 ✓opus

✓ Available
✖ Not available

Figure 1: A quick look at the audio and video formats supported by the system reveals whether any codecs are missing; you can then retroactively install them using Gstreamer plugins.

Listing 1: Installing

```
$ sudo apt install vokoscreen-ng
$ sudo dnf install vokoscreenNG
$ sudo zypper install vokoscreenNG
$ sudo snap install vokoscreen-ng
```



ings can be retroactively edited with an external video editor and uploaded to platforms such as YouTube.

Quickly Installed

Like previous versions, VokoscreenNG is available for both Linux and Windows. The repositories of many distributions currently contain both the old *vokoscreen* (version 2.5.x) and the new *vokoscreen-ng* (version 3.x). When installing via the package manager, make sure you choose the right name and version. The current version is 3.0.3.

On Debian and derivatives, you install VokoscreenNG with the command from the first line of Listing 1. Users with Fedora need the command from the second line. On openSUSE, use the command from Line 3; running the command from Line 4 installs a package in Snap format on your hard disk. An AppImage is already in development [4], but there is no Flatpak thus far.

Format Questions

Before you start, you should think about the formats in which you need the audio and video data for the intended purpose. When you launch VokoscreenNG for the first time, you will first want to click on the tab with the blue info icon on the right in the horizontal bar to see which formats the software supports (Figure 1).

To be able to use as many formats as possible, you will want to install the *gstreamer1.0-plugins-bad*, *gstreamer1.0-plugins-ugly*, and *gstreamer1.0-plugins-libav* libraries (for Debian: *gstreamer1.0-libav*) on your system. On Windows 10 you need a complete codec pack [5].

VokoscreenNG supports the MKV, WEBM, AVI, MP4, and MOV video formats with the x264 and VP8 codecs. Audio formats supported by the program include MP3, FLAC, OPUS, and Vorbis.

Seven Tabs

You can start configuring the settings for a recording in the first tab, in which you specify whether you want to record a full screen, a window, or a freely definable area (Figure 2). If there are several monitors connected to the computer, you can select one of them for full-screen recording. If required, you can also switch on a magnifying glass to highlight certain areas.

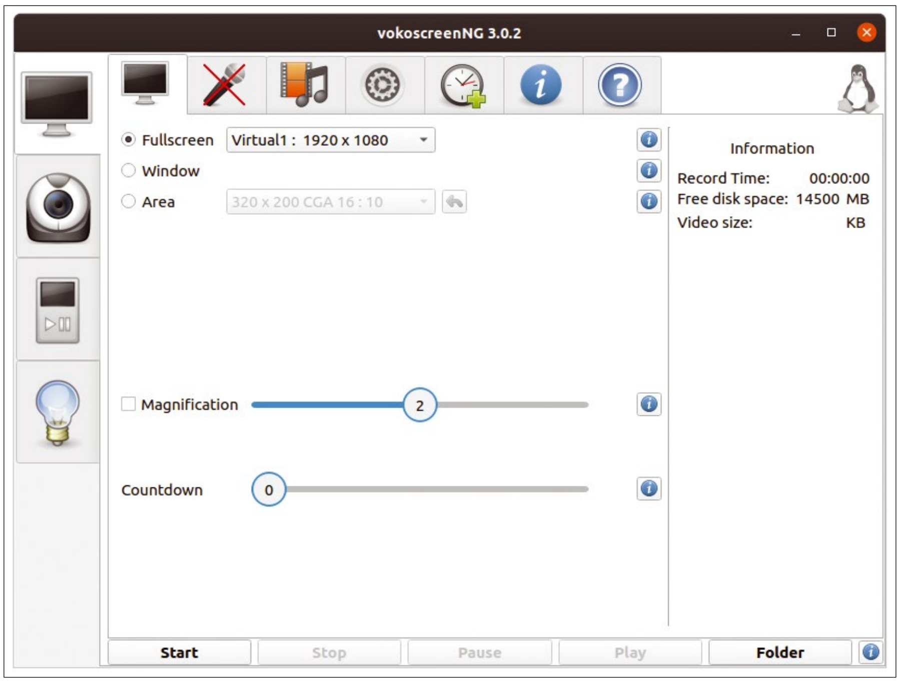
In the tab to the right, you can select an optional webcam and, if needed, a microphone. In the third tab, set the refresh rate for the image, the format for the video, the codecs, and the video and audio data quality (Figure 3). In terms of frame rate, the program supports modern displays with up to 144Hz. Parameters such as the video format, codecs, and above all, the quality have a decisive influence on the size of the resulting video.

In the fourth tab, you can define settings such as minimizing on startup or the behavior of the window during recording (Figure 4). The tab with the dial symbol offers a timer that starts and stops recording at predefined times. The last tab, which has the question mark, includes links to the homepage and source code, as well as the online help and other things.

Two Bars

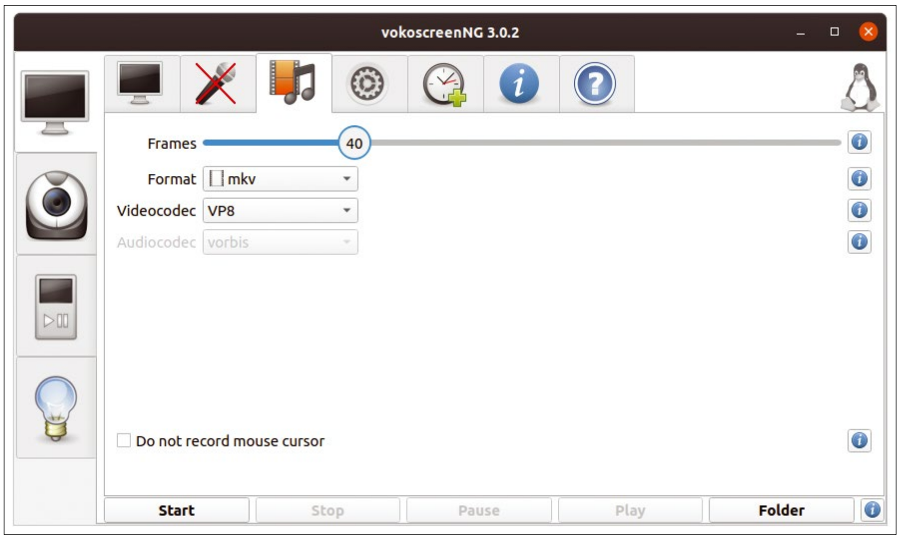
The VokoscreenNG user interface shows a vertical bar with control elements next to the horizontal bar at startup time. Use the vertical bar to connect the system camera or an external camera as a frame within a frame. The webcam table lets you choose one of three selectable sizes for the recording (Figure 5).

The info tab, which might be better described as a log, contains full details of the current session. It also offers the option



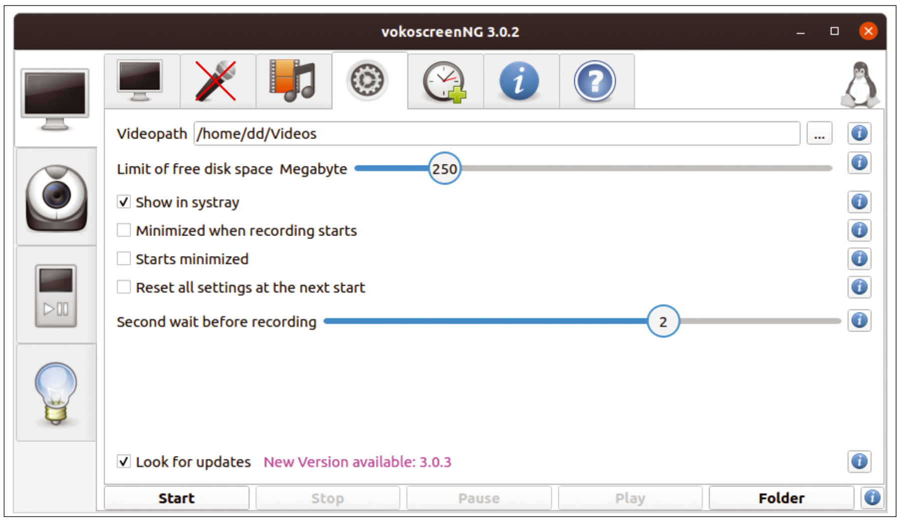
Information
Record Time: 00:00:00
Free disk space: 14500 MB
Video size: KB

Figure 2: Using the tab with the monitor icon in the horizontal bar, you can specify whether you want to capture a window, an area, or the entire screen. In multiscreen environments, you can also specify the monitor you wish to capture.



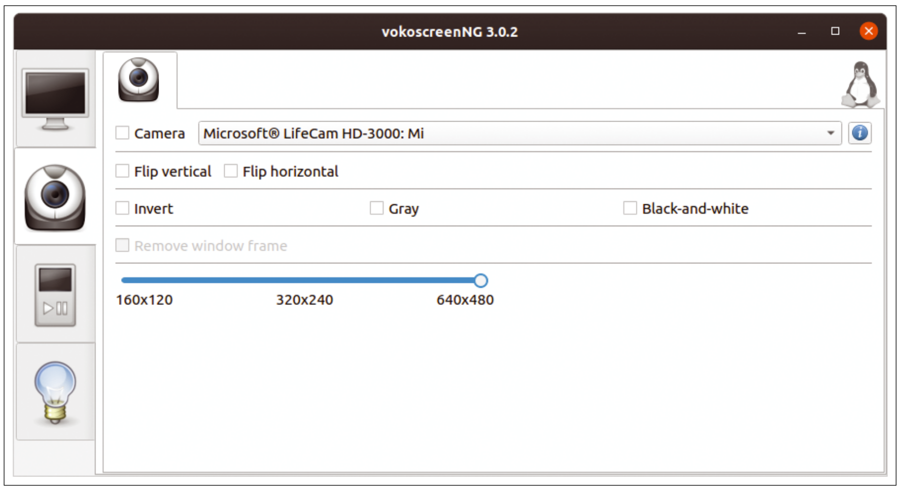
Frames 40
Format mkv
Videocodec VP8
Audiocodec vorbis
Do not record mouse cursor

Figure 3: **In the third horizontal tab, you define the format, codecs, repetition rate, and desired quality. This is also where you decide whether the mouse pointer can appear in the image.**



Videopath /home/dd/Videos
Limit of free disk space Megabyte 250
✓ Show in systray
Minimized when recording starts
Starts minimized
Reset all settings at the next start
Second wait before recording 2
✓ Look for updates New Version available: 3.0.3

Figure 4: **The configuration tab lets you define settings such as the free disk space limit, and how the application behaves at startup time and during recording.**



Camera Microsoft® LifeCam HD-3000: Mi
Flip vertical Flip horizontal
Invert Gray Black-and-white
Remove window frame
160x120 320x240 640x480

Figure 5: **In the first tab of the vertical bar, you select the camera and elect to invert or decolorize the image. If required, you can display the webcam window in three formats up to 640 x 480 pixels.**

of sending data to the developer to help with troubleshooting. In almost all tabs, you will find small info icons on the right-hand side. Clicking on the icon brings up a window with helpful information on the function.

At the bottom of the window, you will see another control bar with five buttons *Start*, *Stop*, *Pause*, *Play*, and *Folder* in all tabs of the horizontal bar. The last button refers to the default directory `$HOME/Video`, where VokoscreenNG stores its recordings.

You can play back the recordings via the fairly simple integrated player or call up a dialog via *Folder* to use an external player. In this case, the globally specified application for the format comes into play.

If desired, you can control VokoscreenNG via keyboard shortcuts. Use Ctrl + Shift + F10 to start a recording, Ctrl + Shift + F11 to stop it, or Ctrl + Shift + F12 to pause it.

Conclusions

VokoscreenNG mainly does what it is supposed to do. A few small inconsistencies are probably due to this being an early version. In multiple-screen environments, the magnifying glass only works on the first screen.

In our lab, it was a bit annoying that the magnifying glass window only opened on the second display and could not be moved to the first display. Getting it to the right place required some practice. The countdown timer also only displayed on the second screen, which caused the program to record the second screen instead of the first. And some of the small info icons do not have any content as yet.

Other than these minor issues, VokoscreenNG handled the basic functions very solidly and did not show any weaknesses in the test. ■■■

Info

[1] VokoscreenNG: *http://www.kohaupt-online.de/hp/*

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

[3] Pipewire: *https://pipewire.org/*

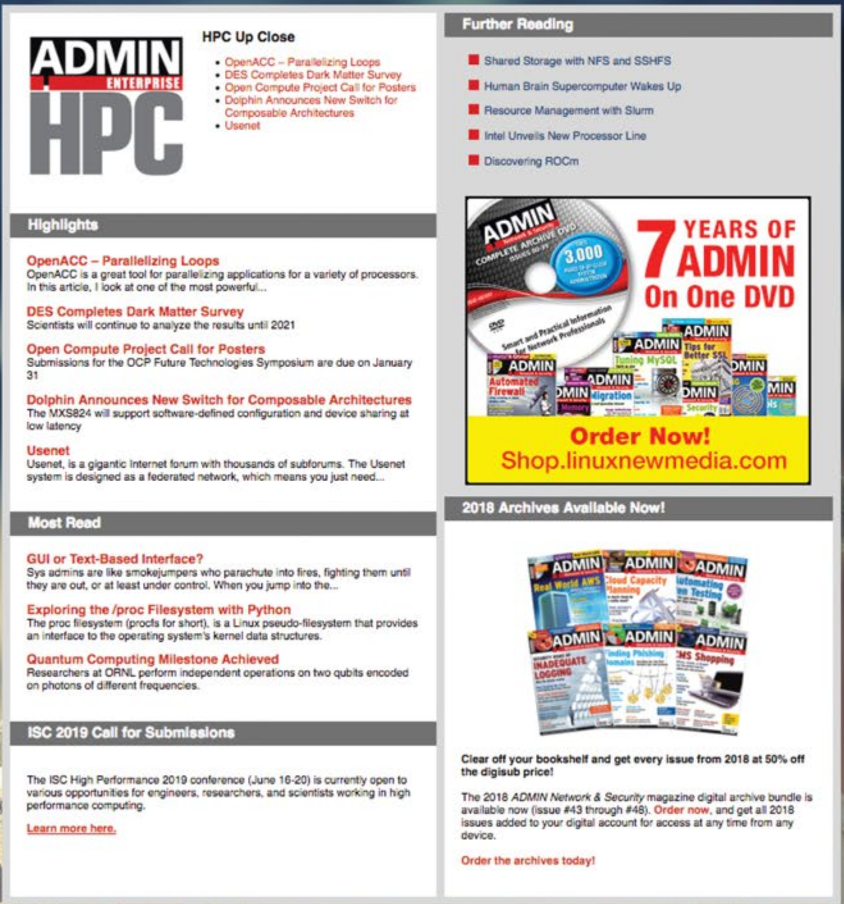
[4] VokoscreenNG with AppImage: *https://github.com/vkohaupt/vokoscreenNG/issues/26*

[5] Windows codecs: *https://www.windows10codecpack.com/*

GOT CLUSTER?

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

https://bit.ly/HPC-ADMIN-Update



ADMIN ENTERPRISE HPC

HPC Up Close

- OpenACC – Parallelizing Loops
- DES Completes Dark Matter Survey
- Open Compute Project Call for Posters
- Dolphin Announces New Switch for Composable Architectures
- Usenet

Highlights

OpenACC – Parallelizing Loops
OpenACC is a great tool for parallelizing applications for a variety of processors. In this article, I look at one of the most powerful...

DES Completes Dark Matter Survey
Scientists will continue to analyze the results until 2021

Open Compute Project Call for Posters
Submissions for the OCP Future Technologies Symposium are due on January 31

Dolphin Announces New Switch for Composable Architectures
The MXS824 will support software-defined configuration and device sharing at low latency

Usenet
Usenet, is a gigantic Internet forum with thousands of subforums. The Usenet system is designed as a federated network, which means you just need...

Most Read

GUI or Text-Based Interface?
Sys admins are like smokejumpers who parachute into fires, fighting them until they are out, or at least under control. When you jump into the...

Exploring the /proc Filesystem with Python
The proc filesystem (procfs for short), is a Linux pseudo-filesystem that provides an interface to the operating system's kernel data structures.

Quantum Computing Milestone Achieved
Researchers at ORNL perform independent operations on two qubits encoded on photons of different frequencies.

ISC 2019 Call for Submissions

The ISC High Performance 2019 conference (June 16-20) is currently open to various opportunities for engineers, researchers, and scientists working in high performance computing.

Learn more here.

Further Reading

- Shared Storage with NFS and SSHFS
- Human Brain Supercomputer Wakes Up
- Resource Management with Slurm
- Intel Unveils New Processor Line
- Discovering ROCm

7 YEARS OF ADMIN On One DVD

Order Now!
Shop.linuxnewmedia.com

2018 Archives Available Now!

Clear off your bookshelf and get every issue from 2018 at 50% off the digisub price!

The 2018 *ADMIN Network & Security* magazine digital archive bundle is available now (issue #43 through #48). Order now, and get all 2018 issues added to your digital account for access at any time from any device.

Order the archives today!

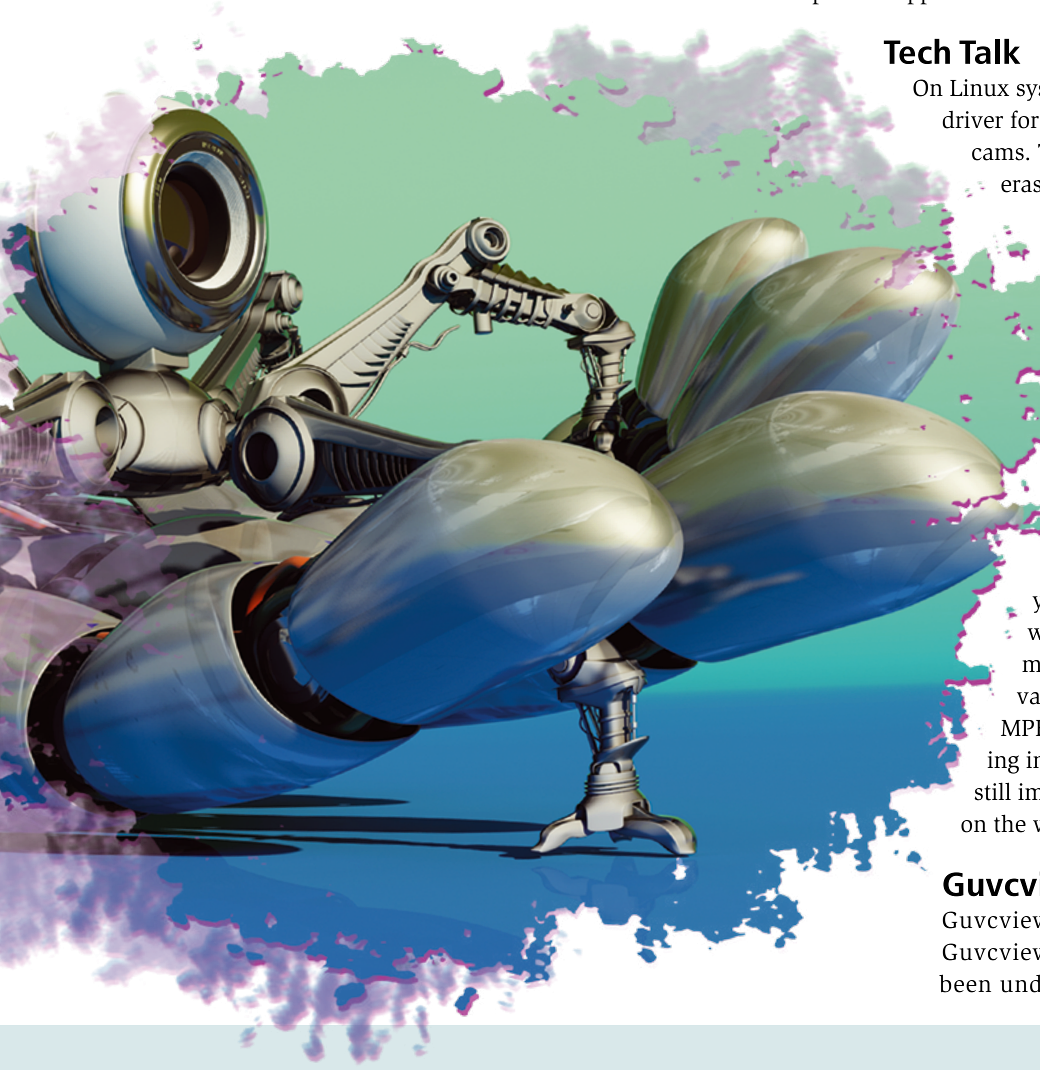
Photo by Kevin Crosby on Unsplash

Webcam streaming with Guvcview and QtCAM

Action!

If your new webcam doesn't work with the default software on your Linux system, try your luck with Guvcview or QtCAM. *By Erik Bärwaldt*

Many webcam manufacturers still don't take Linux seriously, which means that kernel modules for many cameras are often created by freelance developers. In some cases, the manufacturer even changes the chipset during a production run without informing the Linux community. The result of all this uncertainty is that webcam operations in Linux are somewhat unpredictable. A specific system might work out-of-the-box for some cameras but have troubles with others. Users today, however, are accustomed to more seamless hardware configuration. If you, or anyone who depends on you for Linux advice, is having trouble with getting a webcam to work in Linux, one option is to replace the on-board webcam utility shipped with your distro with an alternative application. Guvcview and QtCAM are powerful alternative tools that are suitable for all desktops and support a wide range of cameras.



Tech Talk

On Linux systems, the UVC driver, or the GSPCA driver for older models, can talk to most webcams. The UVC module even supports cameras connected via USB, and it supports webcams built into laptops. A list of compatible webcams can be found on the UVC project's website [1].

The GSPCA driver is used for cameras that do not yet use the UVC driver but can be addressed via a special bridge chipset. A list of supported cameras can be found on the LinuxTV project website [2]. If you use one of these cameras, you can rely on the driver framework, which already contains all additional modules. The modules in turn support a variety of formats, including MPEG2, MPEG4, MJPEG, H.264, and VP8 for moving images, but also JPG, BMP, and PNG for still images. Which format you use depends on the webcam's capabilities.

Guvcview

Guvcview is the GTK+ UVC Viewer [3]. The Guvcview webcam application, which has been under continuous development for more than 10 years, relies on luvcview for video rendering and also makes sound recordings, drawing on PortAudio or PulseAudio for sound capabilities. The software is available from the repositories of popular distributions, and you can download the source code from the project's SourceForge page.

After you install, you will find the Guvcview icon in the desktop menu, usually in the Entertainment Media or Multimedia submenu. After starting the program, two windows open: One window shows the camera image; the other window shows the various settings options.

Use the *Capture Picture* and *Capture Video* buttons to specify whether you will capture a stream or a still image. You can set the recording parameters using the configuration groups located below it in the tabs *Image Controls*, *Video Controls*, and *Audio Controls*.

First of all, adjust the resolution and refresh rate to your requirements in the *Video Controls* area (Figure 1). When the camera resolution changes, the second window with the current camera image is also enlarged or reduced. If you are using several cameras on the computer, select the desired camera via the *Device* selection field.

The *Resolution* selection field offers only those variants for selection that the camera actually supports. In other words, not all resolution options will be displayed for all cameras. Lower resolutions reduce the required storage space.

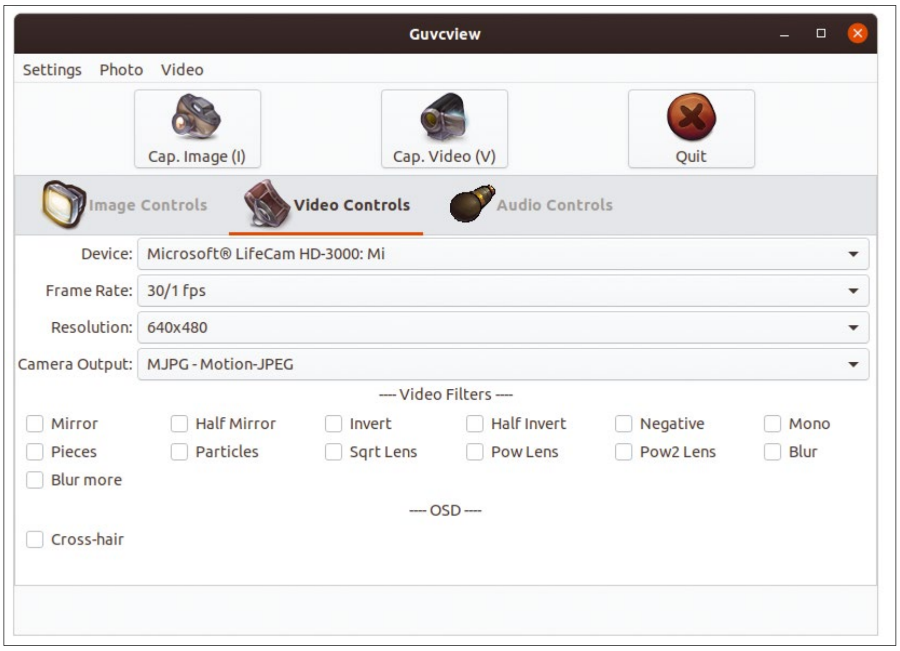
The extensive dialog in the *Image Controls* tab lets you adjust the brightness, contrast, color intensity, and saturation via sliders (Figure 2). You can also adjust the white balance, focus control, backlight correction, and focus.

Since the software transfers changes in real time to the window with the current camera image, you can immediately see how the parameters you set affect the image. This ability to make changes in real time vastly improves the quality of video streams, making incorrect exposure values and faulty coloring a thing of the past.

The third tab, *Audio Controls*, is where you set the parameters for audio recording (Figure 3). Guvcview already comes with some useful presets. You'll find several options in the *Input device* selection field, depending on your choice of computer. Laptops usually come with their own microphones, as do many webcams.

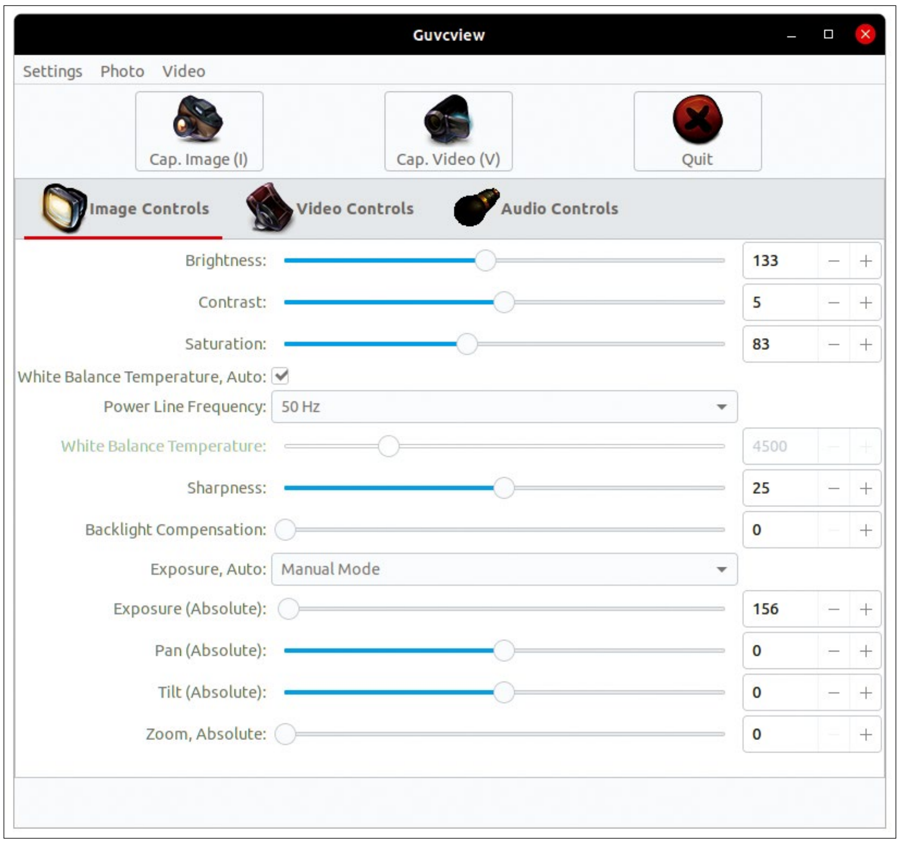
If you want to activate the laptop's webcam microphone instead of the default webcam microphone, you need to specify this explicitly in the selection field. In addition to the sampling rate, this dialog lets you set the number of channels and specify the latencies; if necessary, you can also enable some sound effects such as reverb or echo.

Guvcview lets you select the codecs you want to use. Select the *Video Codec* and *Audio Codec* entries below *Video* in the



Device: Microsoft® LifeCam HD-3000: Mi
Frame Rate: 30/1 fps
Resolution: 640x480
Camera Output: MJPG - Motion-JPEG

Figure 1: Guvcview configuration is based on slide controls and checkboxes.



Brightness: 133
Contrast: 5
Saturation: 83
White Balance Temperature, Auto: ✓
Power Line Frequency: 50 Hz
White Balance Temperature: 4500
Sharpness: 25
Backlight Compensation: 0
Exposure, Auto: Manual Mode
Exposure (Absolute): 156
Pan (Absolute): 0
Tilt (Absolute): 0
Zoom, Absolute: 0

Figure 2: The settings window in Guvcview avoids unnecessary frills.

menubar at the top of the window, and enable the desired codec by checking the radio button.

After selecting the desired codec in *Video Codec Properties* or *Audio Codec Properties*, you have the chance to customize various codec settings (Figure 4).

In a further step, you then define the storage path, the file name, and the container format with which Guvcview will save your content. Use the *File* item in the *Video* and *Photo* menus for this. By default, the software uses the free Matroska format with a file extension of `.mkv`; AVI and WebM are available as alternatives.

If you want to preserve your settings for future sessions, save them in *Settings* | *Save Profile*. *Settings* | *Load profile*, which lets you enable the profile again later on.

When the configuration is complete, start capturing the video or still image by pressing the button at the top left and center of the configuration window. Use the *Quit* button to the right to exit the application; the camera window will also close automatically.

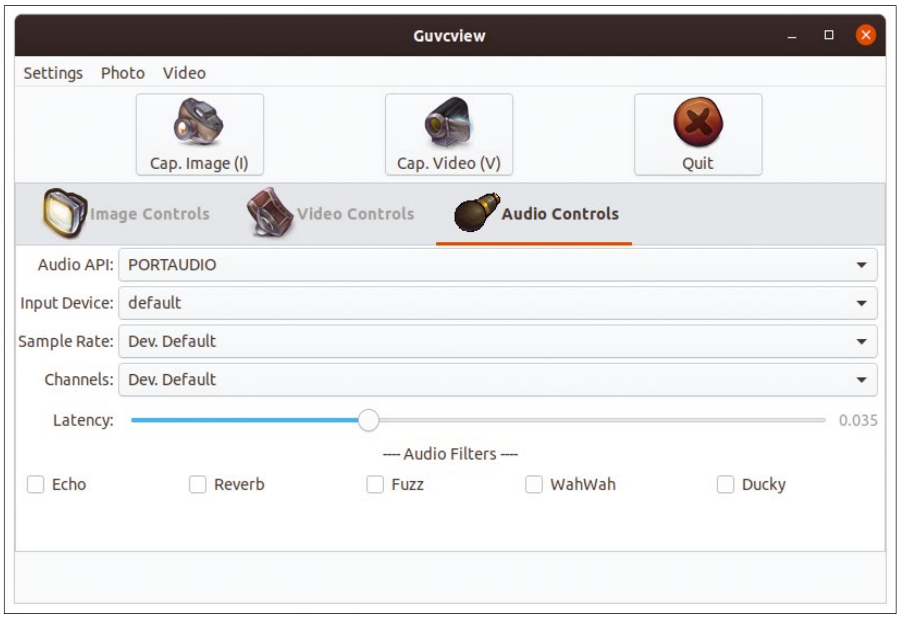
QtCAM

QtCAM, offered by US vendor e-con Systems Inc. and developed in India, is available as free software under the GPLv3 license [4]. Binary packages are available for Ubuntu and its derivatives, Fedora (version 30 and later), and several other popular Linux distros. Documentation and source code are available on the vendor's website.

QtCAM supports numerous e-con cameras, as well as other cameras that can be addressed via the UVC module and V4L2 cameras. The manufacturer provides a list of compatible cameras on its website, but the list does not include many older models.

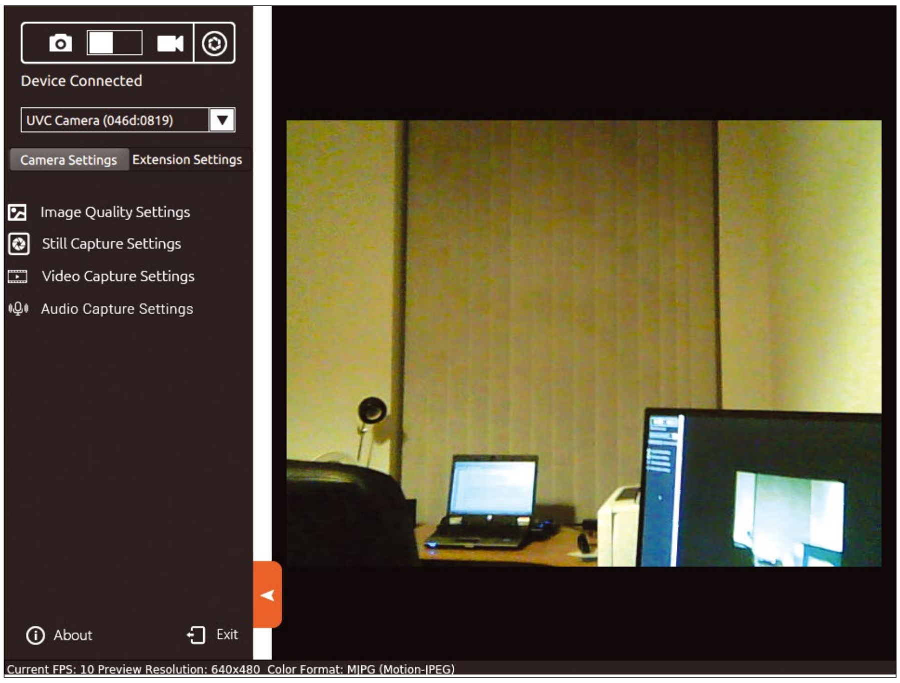
QtCAM offers an intuitive graphical user interface with a state-of-art ergonomic look. You can use up to six cameras simultaneously, which makes the software suitable for video surveillance systems. However, using QtCAM in a mutli-camera, video surveillance system would require some manual operation, because the software does not include surveillance functions such as motion sensors and automatic recording.

The program window is divided into two areas. Vertically, on the left, you will



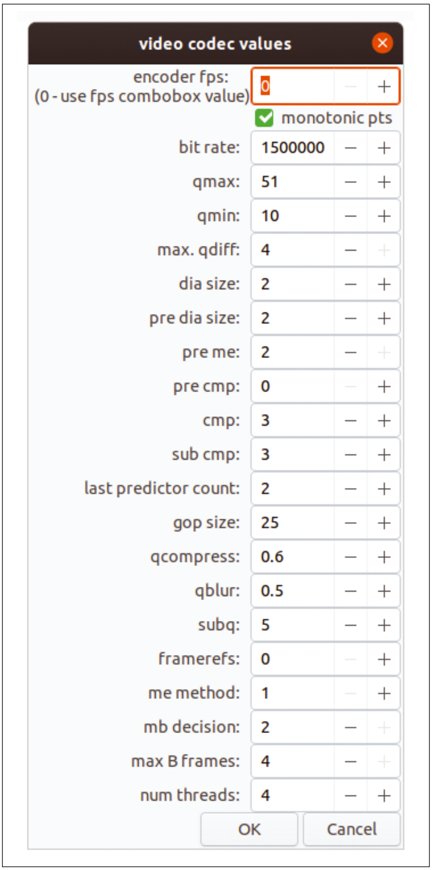
Audio API: PORTAUDIO
Input Device: default
Sample Rate: Dev. Default
Channels: Dev. Default
Latency: 0.035

Figure 3: The audio settings are limited to a few options.



Current FPS: 10 Preview Resolution: 640x480 Color Format: MJPG (Motion-JPEG)

Figure 5: QtCAM uses a single window to control the image display.



encoder fps: (0 - use fps combobox value)	0
✓ monotonic pts	
bit rate:	1500000
qmax:	51
qmin:	10
max. qdiff:	4
dia size:	2
pre dia size:	2
pre me:	2
pre cmp:	0
cmp:	3
sub cmp:	3
last predictor count:	2
gop size:	25
qcompress:	0.6
qblur:	0.5
subq:	5
framerefs:	0
me method:	1
mb decision:	2
max B frames:	4
num threads:	4

Figure 4: Guvcview lets users customize the details of audio and video codecs.

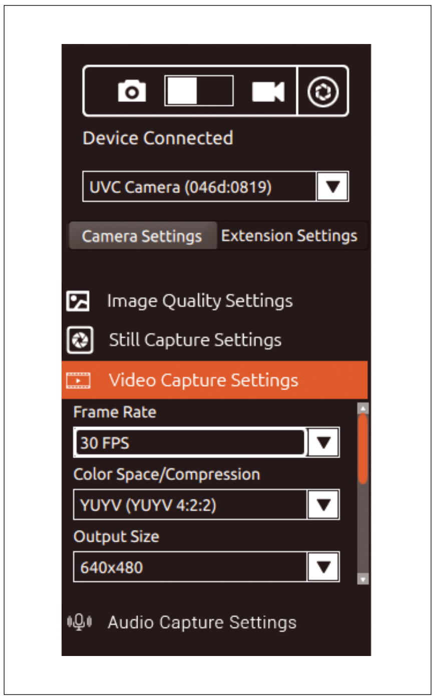

Device Connected
UVC Camera (046d:0819)
Camera Settings | Extension Settings
Image Quality Settings
Still Capture Settings
Video Capture Settings
Frame Rate: 30 FPS
Color Space/Compression: YUYV (YUYV 4:2:2)
Output Size: 640x480
Audio Capture Settings

Figure 6: Although QtCAM offers fewer configuration options than Guvcview, it does allow for device-specific adjustments.

find a setting range that combines different categories into separate settings segments, while the larger area on the right displays the camera images (Figure 5).

After installing QtCAM, you will find an icon in the start menu. Clicking on the icon opens a terminal and prompts you for your admin password (the software requires extended privileges). The program window then opens in full-screen mode.

QtCAM offers very simple user dialogs. In the vertical function bar, you can use the slider at the top to select whether you want to record a still image or a video stream.

To the right of the slider, you will find a trigger that executes the selected function. Below it, select the desired device in a selection box. Depending on the camera model and its technical features, the settings at the bottom vary.

To configure the basic settings, first open the *Image Quality Settings* group. In addition to the brightness, contrast, and color saturation, you can also adjust the white balance and sharpness of the image with separate sliders. Since the software displays the camera image in the right section of the program window while you are making an adjustment, the results of the adjustments are visible on the screen – virtually in real time.

You can use the *Still Capture Settings* category to set special adjustments for still capture, such as the color space and compression. To adjust the resolution, there is also a drop-down list that lists all physical resolutions supported by the camera.

You define the storage path for the recordings in a separate input field. By default, the software uses the `Pictures/` subdirectory in your home folder as the backup path.

In the last step, define the output format of the recordings. You can choose between JPG, BMP, RAW, or PNG. *Video Capture Settings* is where you set the video parameters (Figure 6). The options include the frame rate, color space, and compression specifications, as well as the video resolution.

The software also asks you which output format you want to use. What QtCAM means by this is the container format, but thus far it only supports AVI. In a further step, you can specify the desired encoder. The software lets you choose between MJPEG and H.264, which compresses videos in a better way, resulting in significantly smaller video files of the same quality.

In the last option, set the backup path for the video recordings. The bottom group of settings, *Audio Capture Settings*, is where you configure the audio settings. First select the desired device. You can choose between two devices in the system (e.g., if you use a webcam with an additional microphone on a desktop computer or laptop with a built-in microphone). Depending on the hardware you are using, you then specify the sampling rate and the number of usable recording channels. You can also specify the recording sensitivity.

When the settings are complete, start recording. While a video sequence is running, a green illuminated time display appears top left in the program window. The time display gives you information on how long the recording has been running.

After you press the stop button in the top left corner, the software asks if it should save the new file in the predefined path. QtCAM automatically assigns a file name. To exit the program after saving a file, press *Exit* in the lower left vertical settings bar. The window will close after a prompt.

Conclusions

Guvcview and QtCAM are two powerful webcam applications that facilitate the work of video bloggers and can also capture still images. Gone are the days of overexposed or underexposed shots, where the image resolution was too low or strange hues appeared in the images because the white balance was incorrectly set. You can set up all of these parameters before recording, and you will see the effect of the settings in the preview window. Thanks to Guvcview and QtCAM, you can achieve professional results when working with webcams in Linux. ■■■

Info

[1] Supported UVC cameras: *http://www.ideasonboard.org/uvc/#devices*

[2] Supported GSPCA cameras: *https://www.linuxtv.org/wiki/index.php/Gspca_devices*

[3] Guvcview: *http://guvcview.sourceforge.net/*

[4] QtCAM: *https://www.e-consystems.com/opensource-linux-webcam-software-application.asp*

Author

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

A peek at recent events within some leading Linux distros

Inside View

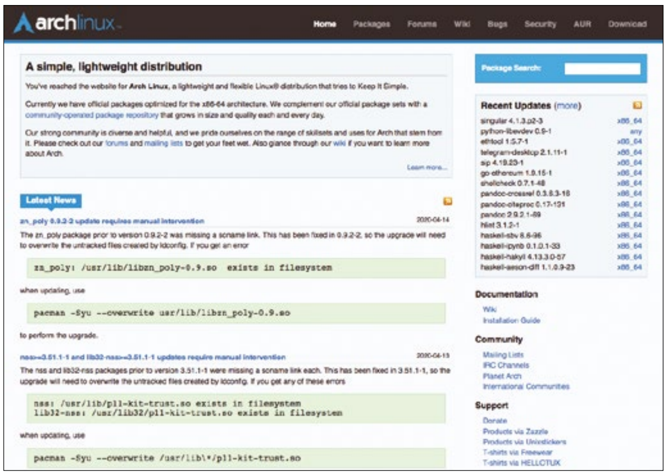
The distro you load on your computer was developed by real people operating through community groups and other organizational structures. This summary of some recent news offers a glimpse into the inner workings of some leading Linux projects. *By Bruce Byfield*

The business of distributions centers around the packaging and testing of applications. However, many distributions have grown so large that their daily activities rival those of Apple and Google. How are distros organized? How are they run? How do they make decisions? What relationships do they have with other free software organizations? This month, Distro Walk touches on a few of these questions through a sample of news items from different distributions. These not only give a glimpse into the complexity of modern distributions, but also show how their daily business might inspire developers to offer their services or help users decide which distribution best suits their philosophy.

Arch Linux

In the early months of 2020, Arch Linux (Figure 1) saw the departure of Aaron Griffin as project leader and the election of Levente Polyak (anthraxx) in a campaign against Gaetan Bisson (vesath), Giancarlo Razzolini (grazzolini), and Sven-Hendrik Haase (svenstaro) [1]. The election was held in accordance with a new process created especially for the election. The process was necessary, because Griffin had been leader since 2007, and Arch has grown and evolved considerably since then.

The process is defined on the project DeveloperWiki [2]. The project leader's role is defined as making decisions when no consensus exists among developers, representing Arch Linux on the board of Software in the Public Interest (the project's nonprofit overseer), representing the project legally, and managing the daily development of the distribution. Eligible voters are recognized Arch developers, trusted users, and staff, who may rank candidates on the ballot. The project leader is elected for two years and may be vetoed by two-thirds of the Arch developers.



archlinux

Home Packages Forums Wiki Bugs Security AUR Download

A simple, lightweight distribution

You've reached the website for **Arch Linux**, a lightweight and flexible Linux® distribution that tries to Keep It Simple.

Currently we have official packages optimized for the x86-64 architecture. We complement our official package sets with a community-operated package repository that grows in size and quality each and every day.

Our strong community is diverse and helpful, and we pride ourselves on the range of skillsets and uses for Arch that stem from it. Please check out our forums and mailing lists to get your feet wet. Also glance through our wiki if you want to learn more about Arch.

Learn more...

Latest News

zn_poly 0.9.2-2 update requires manual intervention 2020-04-14

The zn_poly package prior to version 0.9.2-2 was missing a soname link. This has been fixed in 0.9.2-2, so the upgrade will need to overwrite the untracked files created by ldconfig. If you get an error

```
zn_poly: /usr/lib/libzn_poly-0.9.so  exists in filesystem
```

when updating, use

```
pacman -Syu --overwrite usr/lib/libzn_poly-0.9.so
```

to perform the upgrade.

nss>=3.51.1-1 and lib32-nss>=3.51.1-1 updates require manual intervention 2020-04-13

The nss and lib32-nss packages prior to version 3.51.1-1 were missing a soname link each. This has been fixed in 3.51.1-1, so the upgrade will need to overwrite the untracked files created by ldconfig. If you get any of these errors

```
nss: /usr/lib/p11-kit-trust.so exists in filesystem
lib32-nss: /usr/lib32/p11-kit-trust.so exists in filesystem
```

when updating, use

```
pacman -Syu --overwrite /usr/lib\*/p11-kit-trust.so
```

Package Search:

Recent Updates (more)

singular 4.1.3.p2-3	x86_64
python-libevdev 0.9-1	any
ethtool 1:5.7-1	x86_64
telegram-desktop 2.1.11-1	x86_64
sip 4.19.23-1	x86_64
go-ethereum 1.9.15-1	x86_64
shellcheck 0.7.1-48	x86_64
pandoc-crossref 0.3.6.3-18	x86_64
pandoc-citeproc 0.17-131	x86_64
pandoc 2.9.2.1-69	x86_64
hlint 3.1.2-1	x86_64
haskell-stm 8.6-96	x86_64
haskell-ipynb 0.1.0.1-33	x86_64
haskell-hakyll 4.13.3.0-57	x86_64
haskell-aeson-diff 1.1.0.9-23	x86_64

Documentation
Wiki
Installation Guide

Community
Mailing Lists
IRC Channels
Planet Arch
International Communities

Support
Donate
Products via Zazzle
Products via Unixstickers
T-shirts via Freewear
T-shirts via HELLOTUX

Figure 1: The Arch community distro has a fierce reputation for simplicity and economy of purpose.

Debian

In April, Debian (Figure 2) elected Jonathan Carter as project leader. In a low voter turnout of 339 out of 1,011 eligible developers, Carter defeated Sruthi Chandran and Brian Gupta. He replaced Sam Hartman on April 21 [3].

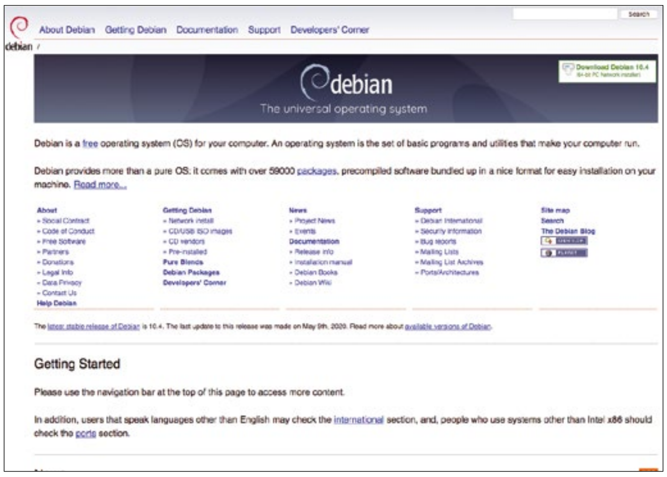
Carter immediately began the old tradition of stating what he had been doing as project leader in his first week and a half, including having a handover session with Hartman, joining the Gnome and Open Source Initiative boards to give Debian representation, writing an annual report for Software in the Public Interest (which also oversees Debian), scheduling future interviews, and weighing in on a community dispute [4].

Meanwhile, the debian-legal mailing list debated whether the Apple Public Source License (APSL) was a free license under the Debian Free Software Guidelines. The issue centered on the *hfsprogs* program, which is needed for the Debian installer to run on Apple PowerBook and Power Mac, and exactly what version of the APSL it is licensed under. [5]. Still another debian-legal thread queried whether the UEFI revocation list could be shipped with Debian as free software [6].

Linux Mint

Clément Lefèbvre, the founder of Linux Mint (Figure 3), writes the distro's

Lead Image © Igor Zakharevich, 123rf.com



Debian is a free operating system (OS) for your computer. An operating system is the set of basic programs and utilities that make your computer run.

Debian provides more than a pure OS: it comes with over 59000 packages, precompiled software bundled up in a nice format for easy installation on your machine. Read more...

The latest stable release of Debian is 10.4. The last update to this release was made on May 9th, 2020. Read more about available versions of Debian.

Getting Started

Please use the navigation bar at the top of this page to access more content.

In addition, users that speak languages other than English may check the international section, and, people who use systems other than Intel x86 should check the ports section.

Figure 2: The vast Debian project selects a new project leader every year by popular election.

monthly blog [7]. Linux Mint is funded by donations. In the interests of transparency, a list of donors is given regularly on the blog. These blogs give a sense of how the project operates.

For example, in March 2020, $9,499 was raised by 527 one-time donors, some 200 of which were repeat donors. Most repeat donors are giving for the 2nd to 15th time, but a handful have donated over 60 times, including one or two over 100 times. Donors come primarily from Europe and the United States, mostly giving $25-$50, although some give as little as $2, and 15 donors gave over $100, including one who gave $200 and one who gave $500.

In March, Linux Mint also had over 100 sponsors, chiefly from Germany and the United States. The amount of each sponsorship was not given, but sponsors appear to be mostly individuals. When companies are mentioned, they vary from toy companies to jewelers, leaving the impression that they are probably mostly individuals as well, differing from donors only in being signed up for regular contributions rather than one-time ones. In addition, 415 patrons also contributed $2,264.

Donations were down slightly in March, compared to $11,301 in February from 621 donors and $14,290 in January from 785 donors. Looking at previous blogs, these appear to be usual seasonal variations.

These figures are worth looking over, because some users prefer to support community-backed distributions. What is worth noting in Linux Mint is the absence of large corporations. Clearly, Linux Mint is a community distribution in the truest sense of the word.

openSUSE

Just as Fedora is the open source face of Red Hat Enterprise Linux, openSUSE (Figure 4) is the open source partner of SUSE. How openSUSE and SUSE interact has been a concern for years, no doubt partly because of SUSE's past changes in corporate ownership. Past efforts to ensure corporation have included joining the Package Hub repository, the SLE Factory First Policy for SUSE employees, and ongoing efforts to make the Public Beta Program more accessible.

In early 2020, Vincent Moutoussamy proposed new initiatives for cooperation. Moutoussamy lists wiki projects to clarify process, efforts to enhance *bugzilla.suse.com*, increased attention to outstanding bugs, and more discussion of the relationship between openSUSE and SUSE. These suggestions amount to an ambitious overhaul of the ongoing relationship between the two distributions [8].

Is your distribution adding something that users should know about? Are your developers doing something interesting on the side? Let us know by writing to edit@linuxpromagazine.com. ■■■

Info

[1] Arch Linux election: *https://www.codetd.com/en/article/9387754*

[2] Arch electoral process: *https://wiki.archlinux.org/index.php/DeveloperWiki:Project_Leader*

[3] Debian election: *https://bits.debian.org/2020/04/results-dpl-elections-2020.html*

[4] Debian Project Leader activities: *https://jonathancarter.org/2020/05/02/free-software-activities-for-2020-04/*

[5] Debian APSL dispute: *https://lists.debian.org/debian-legal/2020/04/threads.html*

[6] Debian UEFI revocation list dispute: *https://lists.debian.org/debian-legal/2020/05/threads.html*

[7] Linux Mint blog: *https://lists.debian.org/debian-legal/2020/04/threads.html*

[8] openSUSE and SUSE: *https://news.opensuse.org/2020/04/29/Discuss-Define-and-be-Transparent-with-the-openSUSE-Community/*

Monthly News – May 2020

This article was posted on Mon, 01 Jun 2020 16:05:05 +0000

Before we start the monthly news, I'd like to thank you for your donations and the support you're giving us. I'd like to thank our partners and sponsors also and everyone who contributes to Linux Mint. Colors Last month we gave a preview of some of the new colors in the Mint-Y theme. We thought [...]

By continuing to use this website, you consent to the use of cookies in accordance with our Cookies Policy.

Figure 3: Linux Mint is funded by donations. A list of donors appears periodically in the Mint blog.

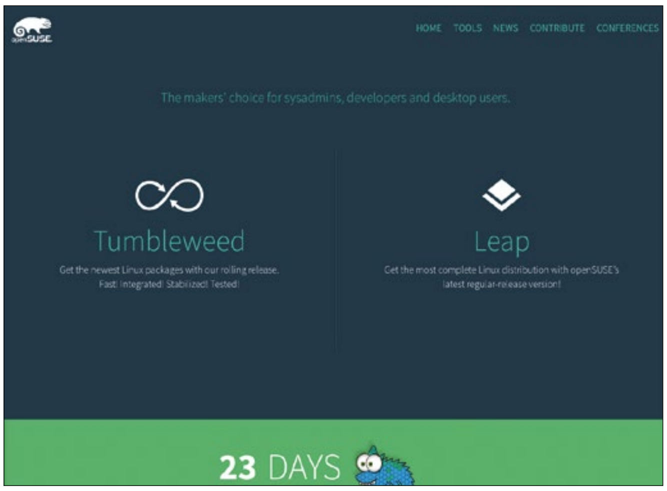

openSUSE
HOME
TOOLS
NEWS
CONTRIBUTE
CONFERENCES
The makers' choice for sysadmins, developers and desktop users.
Tumbleweed
Get the newest Linux packages with our rolling release.
Fast! Integrated! Stabilized! Tested!
Leap
Get the most complete Linux distribution with openSUSE's latest regular-release version!
23 DAYS

Figure 4: OpenSUSE is a community project sponsored by the German IT vendor SUSE.

Adapt the PDFtk PDF tool's call syntax with Go

Tailored

Go is not only suitable for complex server programs, but it also cuts a fine figure with simple command-line tools for automating everyday life. Mike Schilli restructures the signature of a PDF manipulation tool. *By Mike Schilli*

One of my favorite tools at the command line is the PDFtk utility, a veritable Swiss Army knife for merging PDF documents. However, when called, the tool expects unusual syntax for passing parameters, which I find hard to memorize and type every time. That's why I decided to wire up a variant in Go, which tries to guess what the user wants. En route to doing this, the inclined reader will discover how Go reads and writes files, extracts and manipulates single characters from strings, and calls external programs with interactive input, as well as find out how to cross-compile Go programs for various platforms.

For close to a decade now, I've been digitizing paper books and magazines with my scanner and then tossing them into the recycling bin. Sometimes a book comes out as two or more PDFs because the scanner got stuck between two stacks and I had to continue the scanning process with a new document. Sometimes the cover of a hardback book simply will not fit through the scanner feeder, which means that the front and back covers are individual PDF files from a flatbed scanner. PDFtk makes putting the parts together a breeze:

```
$ pdftk book-*.pdf cat output book.pdf
```

What happens here is that the shell grabs any files matching the pattern (`book-*.pdf`) in the current directory, and – if they are numbered `book-1.pdf`, `book-2.pdf`, and so on – passes them to PDFtk in the correct order. The `cat` subcommand tells the tool to stitch together all the input documents in sequence. Finally, PDFtk expects the name of the `output` file after the keyword. So far, so good, but couldn't this all be a little more standards compliant and easier?

Yes, We Can!

The freshly squeezed Go program you are looking at today, aptly named Pdftki, simply grabs the PDF book parts, discovers that they all start with `book-*`, and concatenates them. It decides on `book.pdf` as the name of the target file, as this is the largest common denominator of all the subfiles. And all of this is part of just one simple call:

```
$ pdftki book-*.pdf
```

It's nice, compact, and easy to remember. But what if you need to leave out a page because you have two copies of it, like at the end of `book-1.pdf` and the start of `book-2.pdf`? Thanks to PDFtk, you do this by assigning an uppercase letter to each of the documents and letting the `cat` statement for the second document start at page 2 instead of page 1 (Listing 1) [1].

While PDFtk fully integrates the first file (`1-end`), it skips page one of the second document (`2-end`). This gives you some insight into how powerful PDFtk is, but at a cost of the kind of syntax that has you regularly browsing the man page.

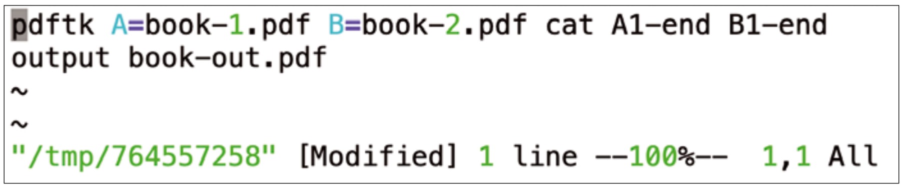
Listing 1: Skipping a Page

```
$ pdftk A=book-1.pdf B=book-2.pdf cat A1-end B2-end output book.pdf
```

Author

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

Lead Image © Sergey Nivens, 123RF.com



```
pdftk A=book-1.pdf B=book-2.pdf cat A1-end B1-end
output book-out.pdf
~
~
"/tmp/764557258" [Modified] 1 line --100%--  1,1 All
```

Figure 1: Calling `pdftki -e` lets the user modify the command in the vi editor before running it.

In contrast to this, the following call to the Go tool Pdftki automatically cobbles together PDFtk parameters for concatenating all parts and assigns them letters with page ranges as shown in Figure 1:

```
$ pdftki -e book-*.pdf
```

The `-e` option then tells it to launch an editor, giving the user the ability to modify the suggested call parameters for PDFtk. After quitting the editor, it then merges the subfiles according to the commands saved. Very handy!

Built-In Help

Listing 2 shows the main program, which uses the `flag` package to interpret its command-line options (such as `-e`, which I just described) and the `Args()` method to extract the list of PDF files specified by the user.

After calling `Parse()` in line 14, the `edit` variable contains a pointer to a `bool` type value. By default, it is set to `false`, but it changes to `true` if the user specifies `-e`. In this case, line 18 starts the `editCmd()` function, which I'll get to later in Listing 4. The user can now modify the arguments determined in line 15 for the PDFtk call in an editor (Figure 1) before line 22 is executed.

The handy *os/exec* package from the Go standard library uses `Run()` to call external programs and their arguments; if so desired, it also captures their standard output and standard error output. Lines 24 and 25 assign `out` buffers of the `Buffer` type from the *bytes* package to the respective attributes. `exec` then runs the command and collects the output in the buffer. If an error occurs, line 29

Listing 2: pdftki.go

```
01 package main
02 
03 import (
04   "bytes"
05   "flag"
06   "fmt"
07   "log"
08   "os/exec"
09 )
10 
11 func main() {
12   var edit = flag.Bool("e", false,
13     "Pop up an editor")
14   flag.Parse()
15   pdftkArgs := pdftkArgs(flag.Args())
16 
17   if *edit {
18     editCmd(&pdftkArgs)
19   }
20 
21   var out bytes.Buffer
22   cmd := exec.Command(pdftkArgs[0],
23     pdftkArgs[1:]...)
24   cmd.Stdout = &out
25   cmd.Stderr = &out
26   err := cmd.Run()
27 
28   if err != nil {
29     log.Fatal(err)
30   }
31 
32   fmt.Printf("OK: [%s]\n", out.
                String())
33 }
```

Listing 3: Calling Help

```
$ ./pdftki -h
Usage of ./pdftki:
  -e    Pop up an editor
```

Listing 4: edit.go

```
package main

import (
  "io/ioutil"
  "log"
  "os"
  "os/exec"
  "strings"
)

func editCmd(args *[]string) {
  tmp, err := ioutil.TempFile("/tmp", "")
  if err != nil {
    log.Fatal(err)
  }
  defer os.Remove(tmp.Name())

  b := []byte(strings.Join(*args, " "))
  err = ioutil.WriteFile(
    tmp.Name(), b, 0644)
  if err != nil {
    panic(err)
  }

  cmd := exec.Command("vi", tmp.Name())
  cmd.Stdout = os.Stdout
  cmd.Stdin = os.Stdin
  cmd.Stderr = os.Stderr
  err = cmd.Run()
  if err != nil {
    panic(err)
  }

  str, err := ioutil.ReadFile(tmp.Name())
  if err != nil {
    panic(err)
  }
  line :=
    strings.TrimSuffix(string(str), "\n")
  *args = strings.Split(line, " ")
}
```

Listing 5: args.go

```
package main

import "fmt"

func pdftkArgs(files []string) []string {
  args := []string{"pdftk"}
  catArgs := []string{}
  letterChr := int('A')

  for idx, file := range files {
    letter := string(letterChr + idx)
    args = append(args,
      fmt.Sprintf("%s=%s", letter, file))
    catArgs = append(catArgs,
      fmt.Sprintf("%s1-end", letter))
  }

  args = append(args, "cat")
  args = append(args, catArgs...)
  args = append(args,
    "output", outfile(files))
  return args
}
```

prints it as a log message. If everything works as intended, line 32 calls the method `out.String()` to print the captured command for the user's perusal.

As an additional goody, the *flag* package provides a simple help function that tells the user which options the program supports if called by typing `pdftki -h` (Listing 3).

Keyboard Pass-Through

Listing 4 comes into play if the user entered the `-e` switch at the command line – that is, if they want to edit the command before executing it.

To call an external program such as an instance of the editor vi, with which the user can also interact, you have to tell the *exec* package to not just pass `Stdout` and `Stderr` from the external program to the identically named channels of the current terminal but also wire up standard input `Stdin`, so that any keystrokes made by the user will actually reach the editor. Go offers matching system file descriptors in the *os* package. Lines 26 to 28 of Listing 4 link the three to the anchor pads of the same name in the *exec* package.

For the user to be able to modify the call in the editor, the `editCmd()` function needs to store the `pdftk` command and its arguments in a file and call the editor with it. After the user has saved the changes and returned, `editCmd()` reads the file and saves its contents in array format in the `args` variable, which was passed in as a pointer.

To do this, `editCmd()` creates a temporary file in the `/tmp` directory. The useful `TempFile()` function from the standard *io/ioutil* package ensures that the new file's name does not collide with files that already exist in `/tmp`, so that different processes won't clobber each other's temp files. After the work is done, the program has to get rid of what is then an obsolete file. This is done by the `defer` call in line 16, which kicks in automatically at the end of the function.

Listing 6: outfile.go

```
package main

import (
  "fmt"
  "path/filepath"
  "strings"
)

func outfile(infiles []string) string {
  if len(infiles) == 0 {
    panic("Cannot have zero infiles")
  }

  ext := filepath.Ext(infiles[0])
  base := longestSubstr(infiles)
  base = strings.TrimSuffix(base, ext)
  base = strings.TrimSuffix(base, "-")

  return fmt.Sprintf(
    "%s-out%s", base, ext)
}

func longestSubstr(all []string) string {
  testIdx := 0
  keepGoing := true

  for keepGoing {
    var c byte

    for _, instring := range all {
      if testIdx >= len(instring) {
        keepGoing = false
        break
      }

      if c == 0 { // uninitialized?
        c = instring[testIdx]
        continue
      }

      if instring[testIdx] != c {
        keepGoing = false
        break
      }

    }
    testIdx++
  }

  if testIdx <= 1 {
    return ""
  }
  return all[0][0 : testIdx-1]
}
```

Reading and Writing

The *ioutil* package also includes the convenience functions `ReadFile()` and `WriteFile()`. They read or write a snippet of text, which must exist as a `byte` array slice (and not as a string), from or to a file.

To do this, line 18 in Listing 4 first uses `Join()` to concatenate the command

and all parameters, separated by space characters, thus creating a long string. The `[]byte()` cast operator then converts the string to a `byte` array slice.

Conversely, `ReadFile()` in line 34 reads the modified file into memory. In line 39, the program converts the resulting bytes in the `line` variable into a character string, using `string()` to do so. It also chops off the line break that vi appends to the end of the file without being asked.

The `Split()` function in line 40 separates the program and its arguments into a new array and assigns it back to the input array, by dereferencing the passed-in pointer. The calling function will then access the modified data instead of the original.

The `pdftkArgs()` function from Listing 5 builds the PDFtk command-line syntax introduced at the beginning of this article; it's preparing the more elaborate parameter set to be used for more complicated cases. It assigns an uppercase letter to each input file and then lists the pages to be joined with `A1-end`, `B1-end`, etc. To do this, it iterates over all input files in line 10 and increments the index `idx` by one, starting at `0`. With this, it increases the ASCII value determined in line 8 by `int('A')` and thus obtains B, C, and so on.

Greatest Common Denominator

That leaves us with the task of determining the name of the output file from all input files using the greatest common denominator. For this purpose, Listing 6 removes the file extension in `outfile()` using the `Ext()` function from the *path/filepath* package. The extension stripped should be `.pdf`.

In `longestSubstr()` from line 23, it then looks, starting at the beginning of the string, for the longest string common to all file names before removing any hyphens in line 17. The call in line 19 adds `-out` to the base name determined in this way, as well as the `.pdf` suffix, which was removed at the beginning. This concludes generating the name of the target file from the source files.

To determine the longest common substring from the beginning, the `longestSubstr()` function starting in line 23 implements a small finite state machine. Line 30 iterates over the names of all input files and stores the currently investigated letter position in the first file name from the list in the variable `c`. The use of zero values eases the control flow here; these are fixed values that Go assigns to variables that have not yet been initialized.

The `byte` type variable `c` is still not initialized after its declaration in line 28. Therefore, according to the Go manual, it has an integer value of `0`. Line 36 uses this to check whether the variable `c` has already been set to the currently investigated letter of the first file name in the inner `for` loop starting in line 30. If not, line 37 goes ahead and primes the variable. After this, `continue` will be ringing in the next round.

During the following passes of the inner loop, line 41 checks whether the currently investigated letter in the next source file from the list still matches the letter from the first file name (and thus the value stored in `c`). When the first mismatch occurs, line 42 sets the variable `keepGoing` to `false`, and line 43 breaks out of the inner loop (`break`). This also causes the outer loop to terminate. The `testIdx` counter is incremented by one in each round to move to the next character, matching as many as possible.

At the end of this, the value of `testIdx` has overshot by one, because the file names start to differ at this index position. Consequently, `longestSubstr()` in line 53 returns an array slice whose highest element index has been reduced by one while running the state machine.

Other Worlds

The `build` command from line 1 of Listing 7 creates the `pdftki` executable for the current platform from the four subprograms. The subsequent call to `pdftki *.pdf` runs the binary with the source files and initiates the desired action. The program does not need any additional modules – just what's available in Go's standard library.

If you want to use the binary under Linux but develop the program on a Mac, you can compile it there with the command from line 2 of Listing 7 and then simply copy the resulting executable to a Linux computer, where it runs without any problems. The opposite route also works, if necessary.

Fast or Maintainable

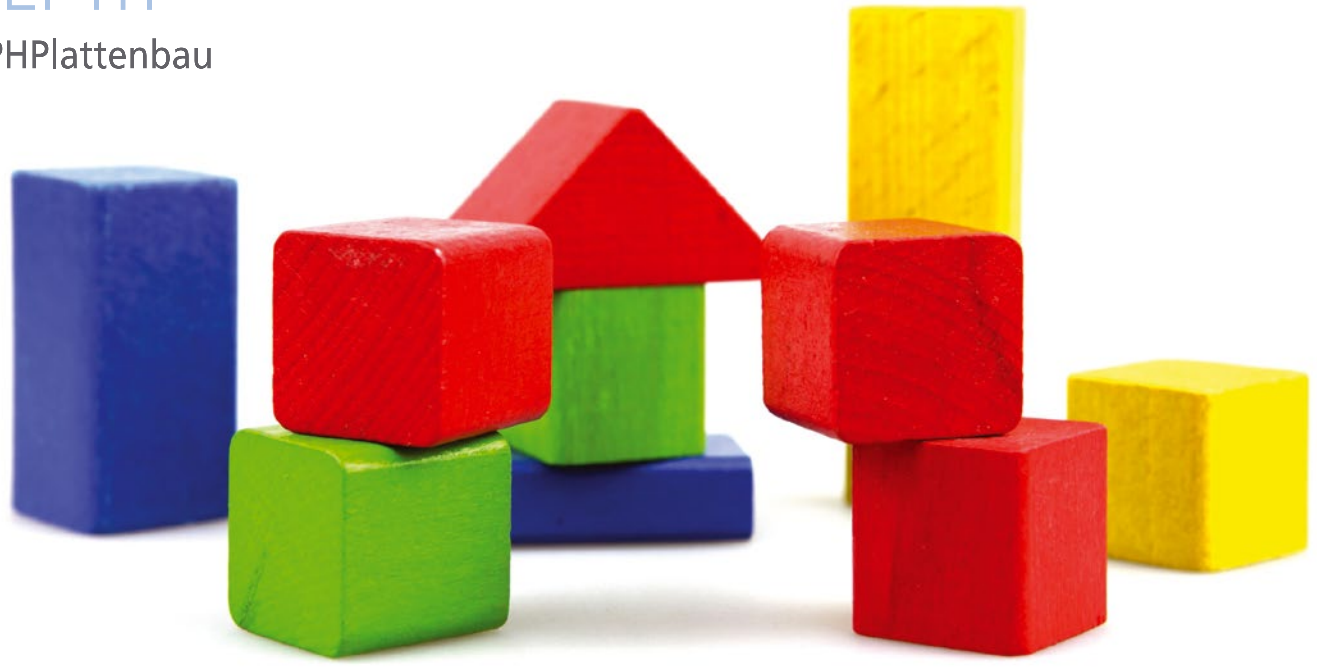
Granted, parts of the Go program I looked at today would have been easier to write as shell scripts. For many routine tasks, a shell hack is practical and often fine for the job at hand. But as the task becomes more challenging – for example, when performing calculations like determining the longest substring from an array – scripts quickly become unwieldy and hard to read. This is where Go benefits from better maintainability, even if the initial overhead is considerably greater. ■■■

Listing 7: Compiling Pdftki

```
01 $ go build pdftki.go edit.go args.go outfile.go
02 $ GOOS=linux GOARCH=i386 go build pdftki.go edit.go args.go outfile.go
```

Info

[1] Listings for this article: *ftp://ftp.linux-magazine.com/pub/listings/linux-magazine.com/237/*



Creating apps with PHP blocks

Build with Blocks

Build simple PHP applications quickly with ready-to-use blocks. *By Dmitri Popov*

PHP is great. It feels approachable and intuitive, and it allows you to write a working application with just a handful of lines of code. PHP basics are relatively easy to master, and there are plenty of books and online resources that can help. As with any programming language, things can get rather hairy sooner or later. But even with basic PHP knowledge, you can build simple yet useful applications. As you progress, you'll notice that many PHP applications share a lot of common code. With time, you may even build your own library of reusable PHP blocks that can significantly reduce the amount of effort required to build an application.

The good news is that you don't have to grow your PHP collection from scratch: The PHPlattenbau project [1] created by yours truly (see the "Personal Note" box) provides several blocks for building a wide range of PHP applications. Each block in PHPlattenbau is actually an application that performs a simple task using one or several common PHP techniques. By adding and mixing these blocks or their parts, you can quickly build a wide range of PHP applications.

Personal Note

I'm not a programmer by any stretch of the imagination. I like writing simple applications and tools that make my everyday life easier. I enjoy the challenge of figuring out how to solve a problem at hand with a few lines of code. Basically, I'm an amateur coder. But there are a great many people like me who are not looking to become professional programmers but who nevertheless would like to be able to write useful applications. These may not be perfect but are good enough to perform the tasks for which they are designed. If you belong to this category, PHPlattenbau is for you.

Tooling Up

You don't need much to get started with PHP and PHPlattenbau on Linux. Obviously, you need PHP 7.x installed on your system. To install PHP on openSUSE, run the command:

```
sudo zypper in php7
```

To do the same on Ubuntu, use:

```
sudo apt install php
```

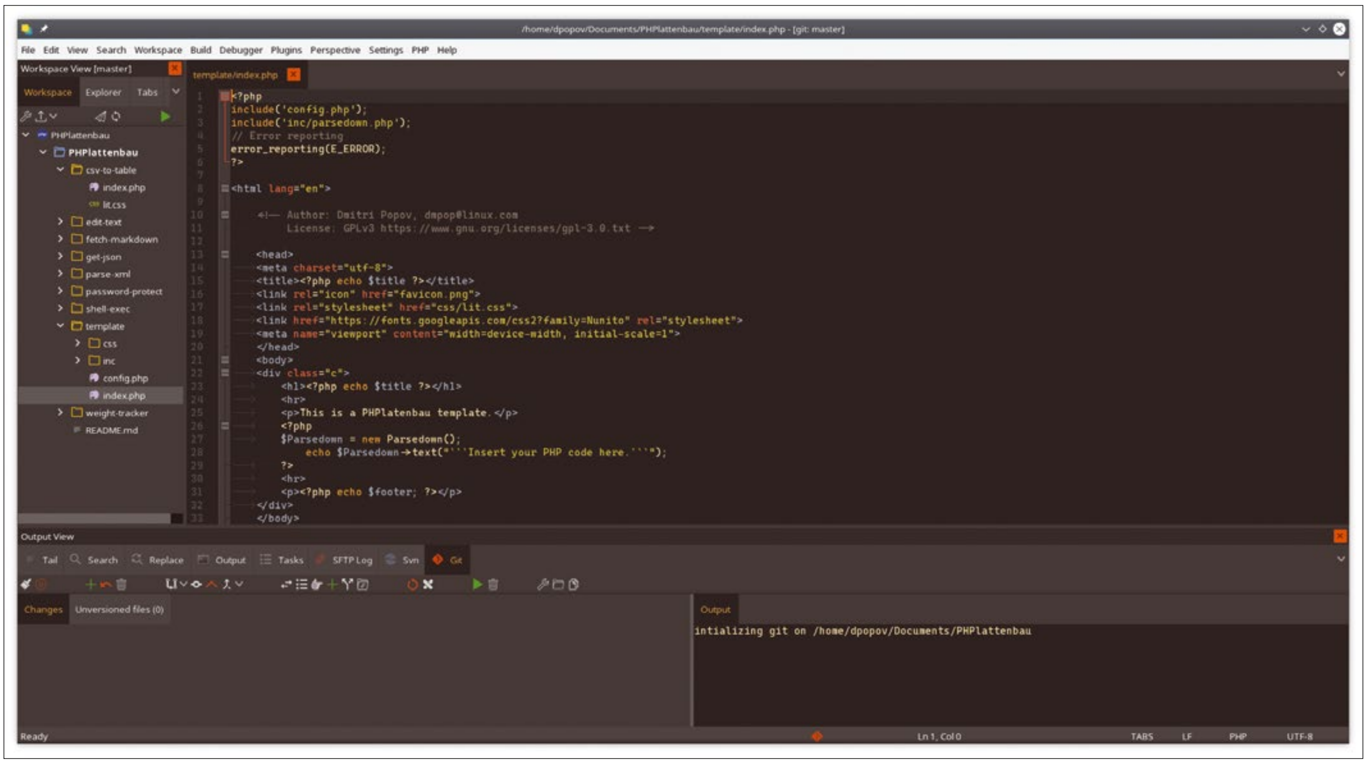
Since PHP comes with a built-in web server, you can use it for testing and experimenting instead of installing a dedicated web server.

You also need a text editor. If you are looking for an editor that is designed specifically for working with PHP and can grow with you, you can do much worse than opting for CodeLite [2] (Figure 1). While you might not need CodeLite's advanced functionality when you start your PHP programming journey, the editor offers plenty of creature comforts that make coding more convenient. This includes workspaces, code folding, autocompletion, bracket matching, and much more. CodeLite is available for all major platforms and mainstream Linux distributions.

Zeal [3], an offline documentation browser, is another utility you might want to add to your toolbox right from the start. Having PHP documentation at your fingertips can be a real timesaver. Instead of trawling the web for answers, you can install Zeal and consult it whenever needed.

Normally, running PHP code requires a dedicated web server like Apache. But to run and experiment with the PHP blocks, you can use PHP's built-in

Lead Image © Vladimir Nenov, 123RF.com



```
<?php
include('config.php');
include('inc/parsedown.php');
// Error reporting
error_reporting(E_ERROR);
?>

<html lang="en">

    <!-- Author: Dmitri Popov, dmpop@linux.com
         License: GPLv3 https://www.gnu.org/licenses/gpl-3.0.txt -->

    <head>
    <meta charset="utf-8">
    <title><?php echo $title ?></title>
    <link rel="icon" href="favicon.png">
    <link rel="stylesheet" href="css/lit.css">
    <link href="https://fonts.googleapis.com/css2?family=Nunito" rel="stylesheet">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    </head>
    <body>
    <div class="c">
        <h1><?php echo $title ?></h1>
        <hr>
        <p>This is a PHPlatenbau template.</p>
        <?php
        $Parsedown = new Parsedown();
            echo $Parsedown->text("```Insert your PHP code here.```");
        ?>
        <hr>
        <p><?php echo $footer; ?></p>
    </div>
    </body>
```

```
intializing git on /home/dpopov/Documents/PHPlattenbau
```

Figure 1: CodeLite is a solid PHP editor.

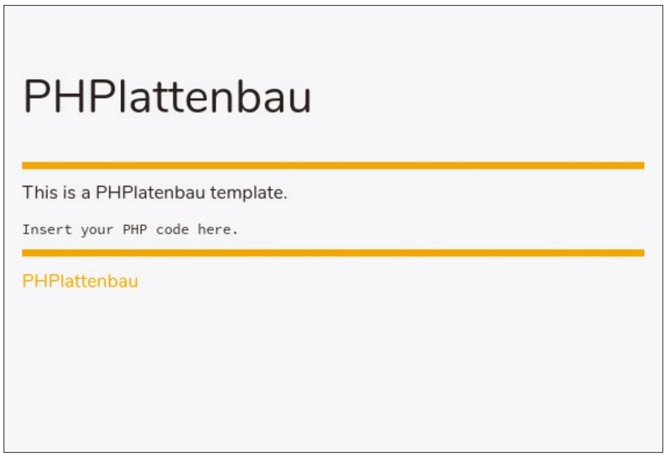
server. In the terminal, switch to the directory of the desired block (e.g., *csv-to-table*) and run either

```
php -S localhost:8000
```

or

```
php -S 0.0.0.0:8000
```

The first command allows you to access the running application only from the same machine the server is running on, while the second command makes it possible to access the application from any machine.



PHPlattenbau

This is a PHPlatenbau template.

`Insert your PHP code here.`

PHPlattenbau

Figure 2: You can use the supplied responsive PHP page template as a starting point for your application.

Block by Block

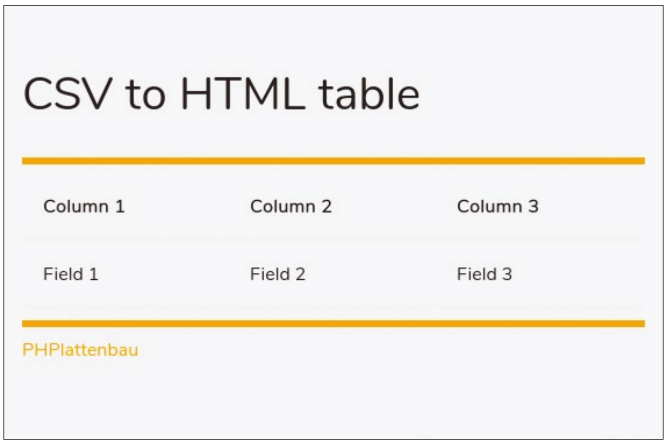
In many programming languages, adding a graphical user interface (GUI) is no trivial matter. With PHP, you can whip up a usable GUI using HTML and a dash of CSS. The template block of PHPlattenbau gives you exactly that: a basic HTML/PHP template that you can use as a starting point for practically any PHP application (Figure 2). The template uses lit [4], a tiny CSS framework, for basic styling. Lit is responsive by design, which means that the template works equally well on desktop and mobile browsers.

Most PHP applications are meant to be run on a web server, so they are exposed to the world. This is not an issue if you have a simple application that only reads and displays data. But if your application includes editing functionality, you might want to limit access to it. One way to do this is to use password protection. This is exactly what the *password-protect* block offers. By using it, you can easily protect any page with a password. To do this, add the following PHP code to the page you want to protect:

```
<?php
include('config.php');
if ($protect) {
    require_once('protect.php');
}
```

Change the default password in the `config.php` file, and you are done. Keep in mind that while this protection mechanism is simple and easy to add, it's not particularly sophisticated. So don't rely on it as your only line of defense.

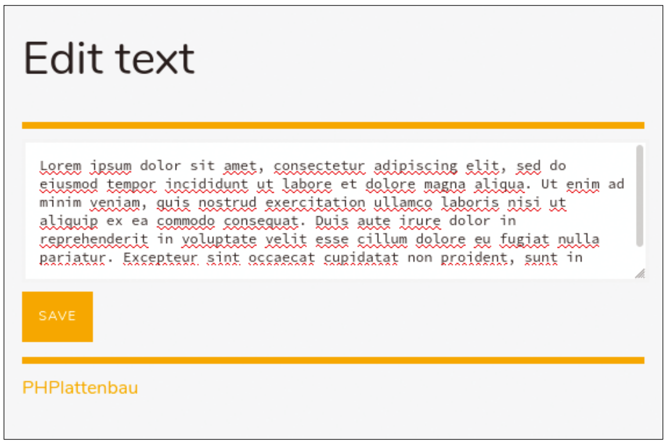
PHP is often used in combination with MySQL or SQLite databases, and there are plenty of tutorials and examples on how to use these database engines as back ends for PHP applications. But using a database for a simple PHP application can be overkill, especially if you consider the amount of effort required. At the very least, you need to write scripts for adding, editing, and querying database records. You also have to ensure that the database queries are properly sanitized (i.e., formatted in a way that prevents misuse). Before you even get that far, you need to deploy a data-


CSV to HTML table

Column 1	Column 2	Column 3
Field 1	Field 2	Field 3

PHPlattenbau

Figure 3: The *csv-to-table* block lets you use a CSV text file as a simple database back end.


Edit text

```
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad
minim veniam, quis nostrud exercitation ullamco laboris nisi ut
aliquip ex ea commodo consequat. Duis aute irure dolor in
reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla
pariatur. Excepteur sint occaecat cupidatat non proident, sunt in
```

PHPlattenbau

Figure 4: The *edit-text* block lets you add text editing capabilities to your PHP applications.

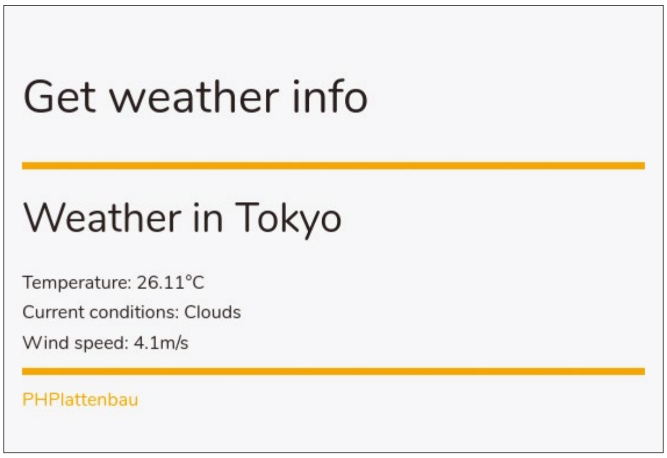
base instance and make sure it works with PHP.

Instead of using a database, you can opt for storing data in comma-separated values (CSV) plain text. Each line in the text file acts as a database record, where each value is separated by a comma (hence the name). The CSV format is easy to read, and you can use a regular text editor or a spreadsheet application to work with CSV data. But how do you view CSV data in a text file using PHP? This is where the *csv-to-table* block comes into the picture. It reads the data in the specified text file (by default, it's `data.csv`) and renders the contents as a nicely-formatted HTML table (Figure 3).

If you need to deal with Markdown in your PHP applications, the *fetch-markdown* block can help. It uses the Parsedown PHP library [5] to parse Markdown-formatted text. In addition to that, the block uses routines to fetch remote contents, which can be useful for publishing content, while delegating the task of editing and managing it to a third-party system.

Speaking of editing, the *edit-text* module allows you to do just that. It's basically a bare-bones editor that you can plug into any PHP application (Figure 4). The block reads the contents of a text file into a text area, where you can edit it. When you press *Save*, the editor creates a backup copy of the text file and saves the modified contents in the file itself.

Many web-based applications nowadays pull data from APIs in XML or JSON format. Two blocks make it possible to add this functionality to your PHP applications: *get-json* and *parse-xml*. As the name suggests, the *get-json* block demonstrates how to pull and parse weather data in the JSON format from the OpenWeatherMap service (Figure 5). It also shows how to obtain values from an HTTP request using the `GET` method. To use the *get-json* block, you need an OpenWeatherMap API key. You can get it free of charge by creating an account with the service. The *parse-xml* block illustrates how to parse and display XML

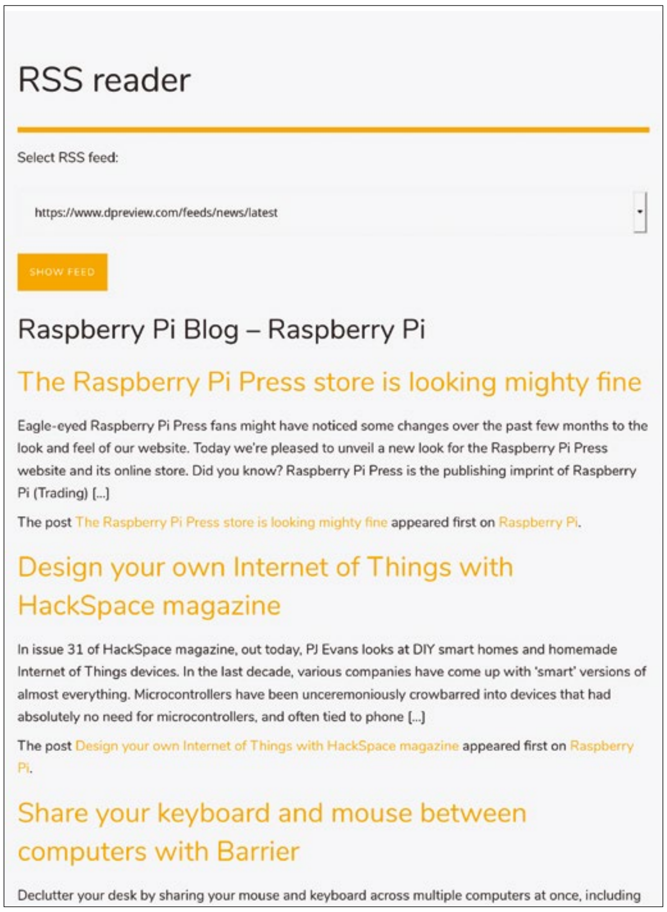

Get weather info

Weather in Tokyo

Temperature: 26.11°C
Current conditions: Clouds
Wind speed: 4.1m/s

PHPlattenbau

Figure 5: Using the *get-json* block, you can fetch and process JSON data.


RSS reader

Select RSS feed:

https://www.dpreview.com/feeds/news/latest

Raspberry Pi Blog – Raspberry Pi

The Raspberry Pi Press store is looking mighty fine

Eagle-eyed Raspberry Pi Press fans might have noticed some changes over the past few months to the look and feel of our website. Today we're pleased to unveil a new look for the Raspberry Pi Press website and its online store. Did you know? Raspberry Pi Press is the publishing imprint of Raspberry Pi (Trading) [...]

The post The Raspberry Pi Press store is looking mighty fine appeared first on Raspberry Pi.

Design your own Internet of Things with HackSpace magazine

In issue 31 of HackSpace magazine, out today, PJ Evans looks at DIY smart homes and homemade Internet of Things devices. In the last decade, various companies have come up with 'smart' versions of almost everything. Microcontrollers have been unceremoniously crowbarred into devices that had absolutely no need for microcontrollers, and often tied to phone [...]

The post Design your own Internet of Things with HackSpace magazine appeared first on Raspberry Pi.

Share your keyboard and mouse between computers with Barrier

Declutter your desk by sharing your mouse and keyboard across multiple computers at once, including

Figure 6: The *parse-xml* block demonstrates how to process XML data.

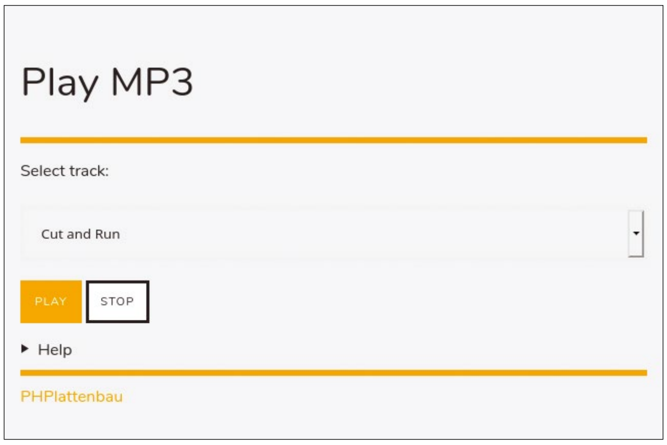
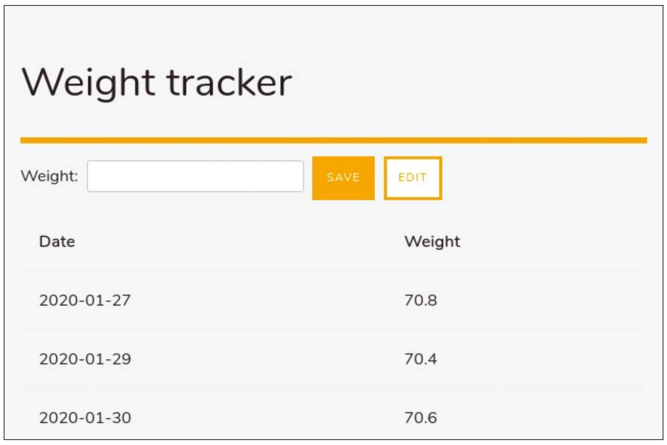


Figure 7: **The *shell-exec* block illustrates how to run shell commands from PHP.**



Date	Weight
2020-01-27	70.8
2020-01-29	70.4
2020-01-30	70.6

Figure 8: **A simple weight tracker application built with PHPlattenbau blocks.**

data by fetching and processing RSS feeds (Figure 6).

Finally, the *shell-exec* block can come in handy when you need to trigger shell commands via PHP. This block scans the `mp3` directory for MP3 files and then populates the drop-down list in the main page with the names of the found files. You can then select the title you like from the drop-down list and use the appropriate buttons to play and to stop the track (Figure 7). This block requires the mpg123 command-line MP3 player to be installed on your system.

Building a Weight Tracker Application

You'll be forgiven for thinking that seven simple PHP blocks and a template are hardly enough to build anything usable. But you really don't need much to create simple yet functional applications. For example, combining the template with the *csv-to-table*, *edit-text*, and *password-protection* blocks, you can whip up a minimal application for tracking weight.

Here's how this application works. The password-protected main page based on the template integrates the *csv-to-table* block that reads CSV data from the `data.csv` file and presents the results as a table. The main page also has a form for entering weight values. When you press the *Save* button, a simple PHP routine gets the value entered in the input field, obtains the current date, and saves both values in the `data.csv` file. You can edit the data file using any text editor, but if you prefer to manipulate the saved data using the application, you can easily integrate the *edit-text* module into it. Add the *Edit* button to the form on the main page, specify the correct name of the data file in the `edit.php` file, and you are done.

The Weight Tracker example application comes with PHPlattenbau (Figure 8), so you can dissect and study it.

Afterword

PHPlattenbau gives you just a few simple PHP blocks. But even this small library lets you create a wide variety of useful applications. Using these blocks, I built several PHP applications that I rely upon in my daily work. This includes the Linkalot bookmark manager [6], the Weather and notes tool for recording weather conditions and notes [7], and the micro.sth publishing application [8]. With a bit of creative thinking, you can put PHPlattenbau to many other practical uses. ■■■

Info

[1] PHPlattenbau: *https://gitlab.com/dmpop/phplattenbau*
[2] CodeLite: *https://codelite.org*
[3] Zeal: *https://zealdocs.org*
[4] lit: *https://github.com/ajusa/lit*
[5] Parsedown: *https://parsedown.org*
[6] Linkalot: *https://gitlab.com/dmpop/linkalot*
[7] Weather and notes: *https://gitlab.com/dmpop/weather-and-notes*
[8] micro.sth: *https://gitlab.com/dmpop/microsth*

Author

Dmitri Popov has been writing exclusively about Linux and open source software for many years, and his articles have appeared in Danish, British, US, German, Spanish, and Russian magazines and websites. Dmitri is an amateur photographer, and he writes about open source photography tools on his Scribbles and Snaps blog at *coffeecode.camera*.

The sys admin's daily grind: rss2email

Sorting the Harvest

In order to keep up to date with security, Charly uses RSS feeds, among other things. He lets rss2email send the most important feeds directly to his mailbox to ensure that nothing is overlooked. *By Charly Kühnast*

RSS feeds are still an essential part of my daily information refueling plan. I sort my feeds in Miniflux, a web-based RSS aggregator, which I wrote about in this column back in 2014 [1]. I am happy to say that Miniflux is still being actively developed.

I have a very small number of RSS feeds emailed to me directly. These are feeds that warn me of acute security problems. By having them delivered directly to my inbox, I can avoid the risk of missing an important article – Miniflux can handle between 200 and 250 feeds, most of which I just skim.

I use rss2email – a very logical choice of name – to email me the feeds. It can be found in the package sources of almost all the popular distributions. On Ubuntu, for example, I installed it with the following call:

```
$ sudo apt install rss2email
```

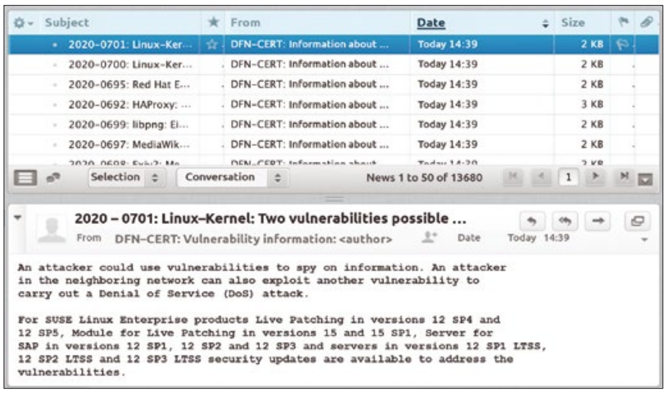
Before using rss2email for the first time, you have to create a configuration file; the approach is very clear-cut. If there is no folder named `.config/` in your home directory, you have to create one and `cd` to it:

```
$ cd
$ mkdir .config
$ cd .config/
```

When you get there, use the editor of your choice to create the `rss2email.cfg` configuration file with the content from Listing 1; the `force-from = True` entry is of special importance here.

Listing 1: rss2email.cfg

```
[DEFAULT]
from = rss@mydomain.com
force-from = True
html-mail = False
to = charly@mydomain.com
```



```
Subject                   From                            Date          Size
• 2020-0701: Linux-Ker...  DFN-CERT: Information about ...  Today 14:39   2 KB
• 2020-0700: Linux-Ker...  DFN-CERT: Information about ...  Today 14:39   2 KB
• 2020-0695: Red Hat E...  DFN-CERT: Information about ...  Today 14:39   2 KB
• 2020-0692: HAProxy: ...  DFN-CERT: Information about ...  Today 14:39   3 KB
• 2020-0699: libpng: Ei... DFN-CERT: Information about ...  Today 14:39   2 KB
• 2020-0697: MediaWik...   DFN-CERT: Information about ...  Today 14:39   2 KB

News 1 to 50 of 13680

2020 – 0701: Linux–Kernel: Two vulnerabilities possible ...
From DFN–CERT: Vulnerability information: <author>    Date Today 14:39

An attacker could use vulnerabilities to spy on information. An attacker
in the neighboring network can also exploit another vulnerability to
carry out a Denial of Service (DoS) attack.

For SUSE Linux Enterprise products Live Patching in versions 12 SP4 and
12 SP5, Module for Live Patching in versions 15 and 15 SP1, Server for
SAP in versions 12 SP1, 12 SP2 and 12 SP3 and servers in versions 12 SP1 LTSS,
12 SP2 LTSS and 12 SP3 LTSS security updates are available to address the
vulnerabilities.
```

Figure 1: Charly uses rss2email to deliver information from RSS feeds about acute vulnerabilities directly to his mailbox.

Listing 2: Enter and check RSS feed

```
01 $ r2e add DFNCert https://adv-archiv.dfn-cert.de/rss/latest
02 $ r2e list
03 0: [*] DFNCERT (https://adv-archiv.dfn-cert.de/rss/latest -> charly@mydomain.com)
```

Listing 3: Cron job for rss2email

```
*/30 * * * * /usr/bin/r2e run > /dev/null 2>&1
```

Normally, rss2email would use a sender address from the RSS feed if it found one there. However, on many mail systems, this leads to the spam filter rightly becoming suspicious. The `force-from = True` entry now causes rss2email to always use the address stored in the `from` line. Whether you set `html-mail` to `True` or `False` is a matter of taste.

Now it's time to choose one or more RSS feeds that you want rss2email to check out regularly. The syntax for this is:

```
r2e add <title> <URL>
```

For the DFN Certificate RSS feed, which informs admins of current security problems, look at the first line in Listing 2. Whether the `add` action worked can be verified using `r2e list` (line 2). It worked in my example.

The `r2e run` command traverses the feeds and dispatches the emails. Now all I need to do is write a crontab entry to run the command at specified intervals (e.g., every 30 minutes). To do so, I type `crontab -e` and add the crontab line from Listing 3.

Sure enough, the first messages start to slowly arrive in my mailbox (Figure 1). The important thing here is to be careful about the rss2email feeds you choose; otherwise you might flood your inbox – just staying with Miniflux would be preferable to that. ■■■

Info

[1] "The sys admin's daily grind: Miniflux" by Charly Kühnast, *Linux Magazine*, issue 164, July 2014, p. 50, *https://www.linux-magazine.com/Issues/2014/164/Charly-s-Column-Miniflux/(language)/eng-US*

Author

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

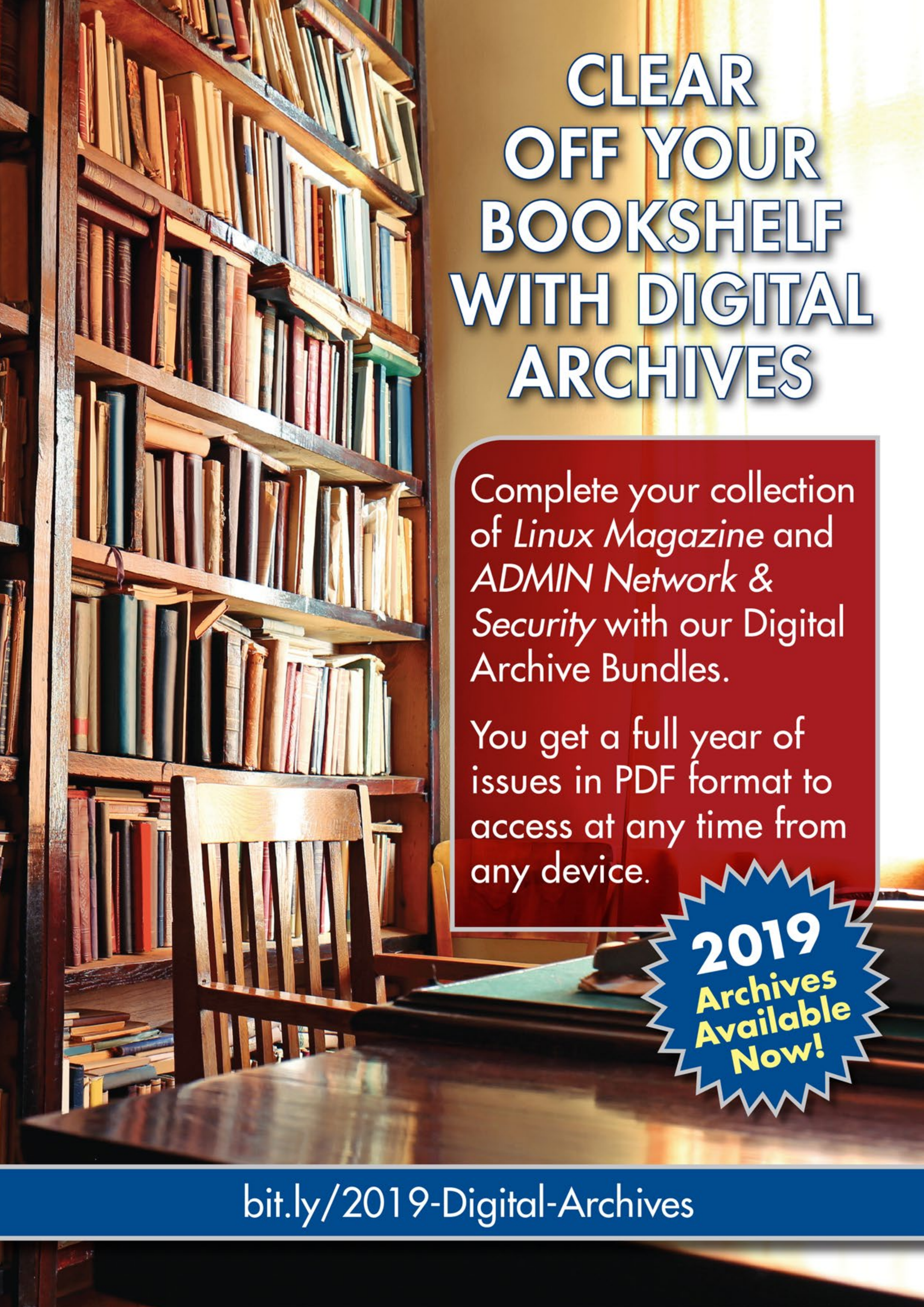
CLEAR OFF YOUR BOOKSHELF WITH DIGITAL ARCHIVES

Complete your collection of *Linux Magazine* and *ADMIN Network & Security* with our Digital Archive Bundles.

You get a full year of issues in PDF format to access at any time from any device.

2019 Archives Available Now!

bit.ly/2019-Digital-Archives



Client-free remote desktop

Holy Guacamole!

Use Apache Guacamole to connect to remote servers from within a web browser. *By Mayank Sharma*

Apache Guacamole [1] is billed as a clientless HTML5 web application that you can use to access your remote servers and desktops. It's called clientless, because Guacamole only requires a web browser unlike other remote desktop solutions that require a client to communicate with the server.

Although Guacamole only reached the 1.0 milestone (the latest version is 1.1) in 2019, the project has been in development for about a decade. Despite its low version number, the project has a mature code base, which is malleable enough to fit all kinds of deployments. You can use it in simple standalone local networks and also on enterprise networks where it can integrate with other existing resources to enhance security and user management.

Guacamole supports all the popular remote desktop protocols including VNC, RDP, SSH, and Telnet. The most recent addition to the list is a Kubernetes client that you can use to attach to the console inside a container. In addition to its protocol support, Guacamole has several enterprise integration capabilities, including LDAP authentication, Duo two-factor authentication (2FA), TOTP 2FA, CAS authentication, OpenID Connect authentication, HTTP header authentication, and more.

The Guacamole client/server architecture consists of a client-side layer implemented in HTML and JavaScript. This browser front-end client layer communicates with the Tomcat Java-based servlet container. The server layer is chiefly exposed as the `guacd` proxy daemon.

Get Guacamole

Installing Guacamole is an involved process. You must fetch various dependencies before you download and compile the latest version from source. The process is time-consuming but well-documented [2]. Alternatively, you can also install Guacamole via Docker. Again the project has clear step-by-step documentation [3] if you prefer to take this route.

But if all this sounds like too much work, you can also use a script that will fetch all the dependencies, configure all the components, and leave you with a working installation with minimal intervention on your part.

Ubuntu users can use the *guac-install* script [4] to get a working installation of Guacamole. If you have a CentOS 7 or a RHEL 7 installation, you can use the *guacamole-install-rhel* script [5].

Both scripts go about the task slightly differently. By default, the Ubuntu script will only prompt you for the database's password. It does however ship with several options that you can use to override the default behavior. For instance, the `--totp` switch will enable 2FA. You can also use the script to upgrade your installation whenever there is a new Guacamole version.

The RHEL/CentOS script is more verbose. It guides you through a few interactive menus that ask for information to help set up Guacamole as per your requirements. The most notable of these is the SSL certificate type menu that gives you the option to either install a Let's Encrypt certificate, a self-signed one, or none at all. If you don't install a certificate, Guacamole will operate over an unencrypted HTTP connection. You should only use this if you're using a certificate from another authority besides Let's Encrypt. Even for small networks, or even while evaluating Guacamole, I'd suggest you use the self-signed option instead of no encryption at all.

Irrespective of which script you choose, you will have a working Guacamole installation in no time. Again, my advice would be to test the scripts inside

Photo by Rafael Arizaga on Unsplash

the safe confines of a virtual machine before deploying them on a physical server.

Regular Housekeeping

When the scripts have completed the installation, they'll both show you the URL for accessing the Guacamole interface. Usually it is either *http://localhost:8080/guacamole* or *https://localhost:8443/guacamole*. You can substitute `localhost` with the IP address of the machine on which you've run the script in order to access it from any other machine on the network.

This should bring up the login page. You can log in using the default credentials, which are *guacadmin:guacadmin*. You are now at the rather bland-looking dashboard since you haven't added any remote desktop connections yet.

Before doing that, the first order of business is to change the default password. To do so, click on the username in the top-right corner of the dashboard and select *Settings* from the drop-down menu. Next, jump to the *Preferences* tab and scroll down to the *Change Password* section to update the password.

While you are here, take some time to explore the other settings. Make sure Guacamole has the right time zone, for instance. Multilingual users can also use the *Display language* field to select another supported language, such as Dutch, Spanish, French, Italian, and more.

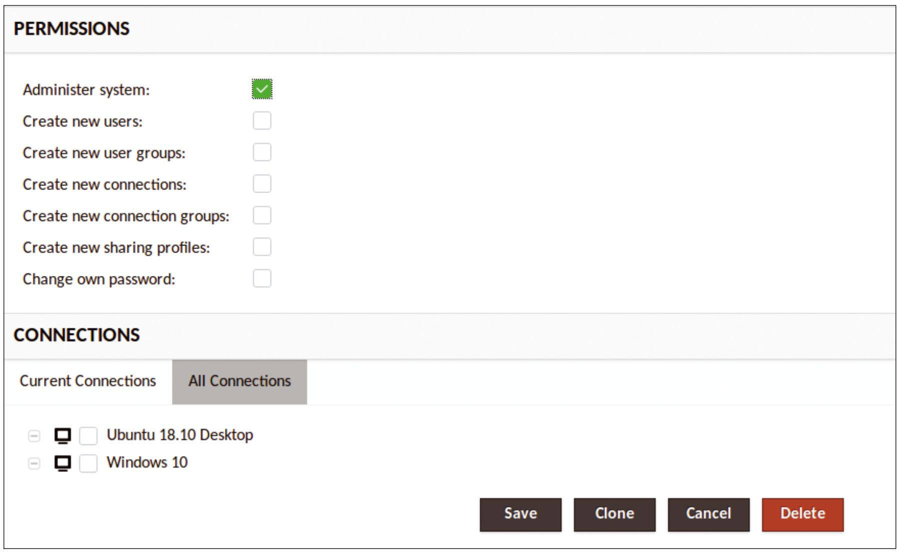
Once you are comfortable with Guacamole, you can return to the Settings screen and add more users from under the *Users* tab. Guacamole has useful permission settings (Figure 1) that you can access from this interface. You can give users the ability to become full administrators or restrict their administrative abilities. For instance, regular non-administrative users can only access the remote machines for which you've granted permission. But you can give them some administrative powers, such as the ability to add new users or create new connections.

Make Connections

Now you are all set to create a new remote desktop connection. Log into the Guacamole dashboard and head to *Settings* | *Connections* and click the *New Connection* button. This opens the Edit Connection page where you can enter various details to describe your connection (Figure 2). Remember that not all options are necessary to establish a connection. You can begin with the least number of details and then fine-tune the connection as per your requirements, once it has been established.

Also note that the requested parameters change depending on the protocol you use for connecting to the remote desktop. Begin by adding a VNC connection to a remote Linux machine on the network. First make sure there's a VNC server running on the remote machine (see the box "Set Up TightVNC Server").

Start by adding a name for the connection, such as *Ubuntu 18.10 Desktop*. Leave the *Location* as ROOT and select *VNC* from the *Protocol* drop-down menu. You can then skip over the next few sec-



PERMISSIONS

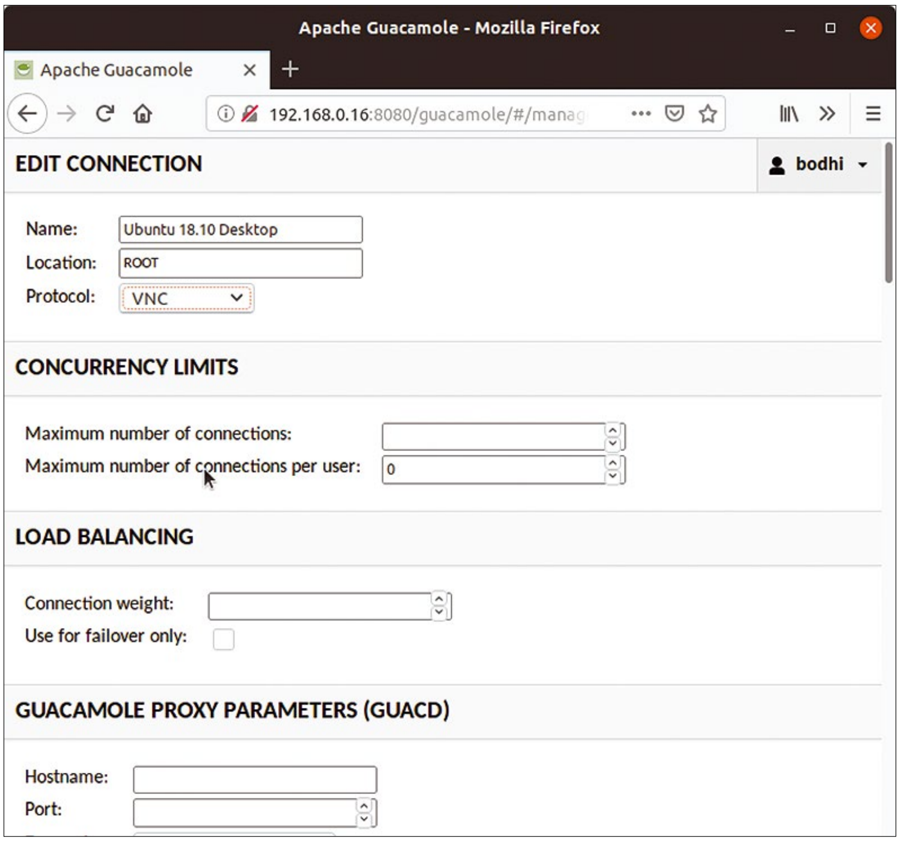
Administer system:	✓
Create new users:	☐
Create new user groups:	☐
Create new connections:	☐
Create new connection groups:	☐
Create new sharing profiles:	☐
Change own password:	☐

CONNECTIONS

Current Connections | All Connections

- ☐ Ubuntu 18.10 Desktop
- ☐ Windows 10

Figure 1: Unless explicitly allowed, users won't be able to change their own passwords by default.



Apache Guacamole - Mozilla Firefox
Apache Guacamole
192.168.0.16:8080/guacamole/#/manag
bodhi

EDIT CONNECTION

Name:	Ubuntu 18.10 Desktop
Location:	ROOT
Protocol:	VNC

CONCURRENCY LIMITS

Maximum number of connections:	
Maximum number of connections per user:	0

LOAD BALANCING

Connection weight:	
Use for failover only:	☐

GUACAMOLE PROXY PARAMETERS (GUACD)

Hostname:	
Port:	

Figure 2: Images transmitted to Guacamole over VNC are always encoded losslessly as PNG images.

Set Up TightVNC Server

There are several VNC servers, but I prefer using the cross-platform open source TightVNC Server that is available in the official repositories of the mainstream distributions. You can install it on Ubuntu with:

```
sudo apt install tightvncserver
```

On Fedora, use:

```
sudo dnf install tightvncserver
```

Then run `vncserver` to set it up. You'll be prompted to enter and verify a password to access your machine remotely. You'll also have the option to create a view-only password. The process then creates the necessary configuration files and connection information for the server.

Next, install the lightweight Xfce 4 desktop environment, which is what I prefer to use to power my remote desktop connection. On Ubuntu, do this with:

```
sudo apt install xfce4 xfce4-goodies
```

On Fedora, install Xfce 4 with:

```
sudo dnf install xfce4 xfce4-goodies
```

Now you need to tell your VNC server to bring up the Xfce 4 desktop whenever it detects an incoming connection request. For this, you need to edit the `~/.vnc/xstartup` file:

```
$ nano ~/.vnc/xstartup

#!/bin/bash
xrdb $HOME/.Xresources
startxfce4 &
```

The commands in the file first ask the VNC server to read the user's `.Xresources` file, which is where users specify changes to certain graphical desktop settings, such as terminal colors. The VNC server will then launch the Xfce 4 desktop.

Then make the file executable to ensure that the VNC server will be able to use this new startup file properly:

```
$ sudo chmod +x ~/.vnc/xstartup
```

Finally, restart the VNC server.

```
$ vncserver
New 'X' desktop is dholak:1

Starting applications specified in /home/bodhi/.vnc/xstartup
Log file is /home/bodhi/.vnc/dholak:1.log
```

The server is now all set to answer VNC connection requests.

tions for the time being and jump straight to the *Parameters* section. Here, enter the hostname or the IP address of the remote machine you wish to access, along with the port number. For a VNC connection, the port number is 5900 plus the display the session is running on. So if your VNC server is running on `:1`, enter *5901* in the *Port* parameter. Finally, enter the password for accessing the VNC session that you specified while setting up TightVNC Server, in the *Authentication* field.

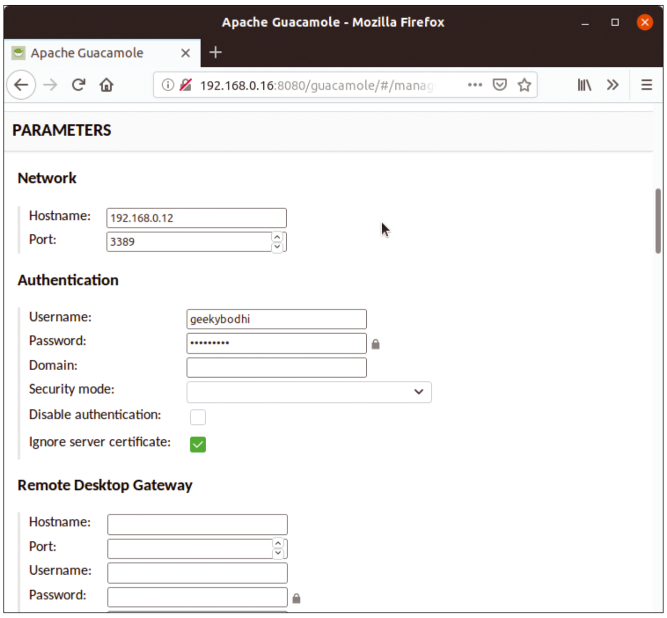
That's all there is to it. Scroll down to the bottom of the page and click the *Save* button. The newly added connection will now be listed on the dashboard. Double-click on the connection to remotely access the Xfce desktop on the Ubuntu installation.

Window to Windows

You can create other connections by selecting different protocols such as RDP.

Again, head back to *Settings* | *Connections* and click the *New Connection* button. Here, give your connection a name and select RDP from the *Protocol* pull-down. Scroll down to the *Parameters* section and enter the hostname or the IP address of the Windows machine you wish to access and enter *3389* in the *Port* field. And just like before, enter the authentication information for the Windows user you wish to remotely access.

You can now save the details and try connecting to the Windows machine. Depending on the version of Windows you want to access, you might have to take additional steps. First up, in any case, you'll have to enable Remote Desktop Sharing in Windows. For this, switch to the Windows machine and head to *Control Panel* | *System and Security* and select the *Allow remote access* option


Apache Guacamole - Mozilla Firefox
192.168.0.16:8080/guacamole/#/manag

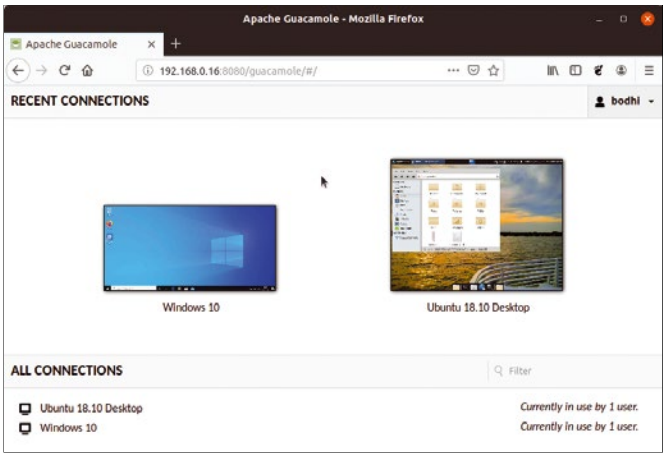
PARAMETERS

Network
Hostname: 192.168.0.12
Port: 3389

Authentication
Username: geekybodhi
Password: •••••••••
Domain:
Security mode:
Disable authentication: ☐
Ignore server certificate: ✓

Remote Desktop Gateway
Hostname:
Port:
Username:
Password:

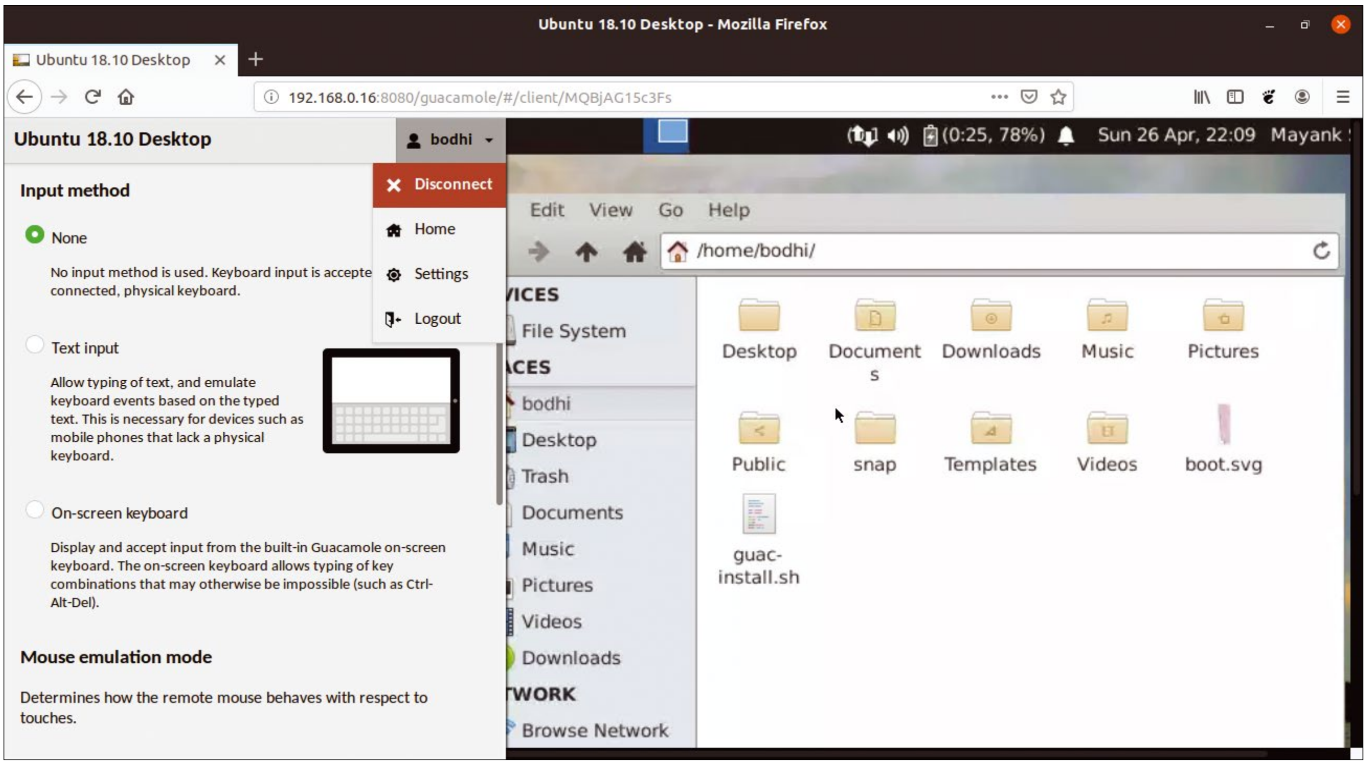
Figure 3: By default, Guacamole selects a security mode as part of the negotiation process.


Apache Guacamole - Mozilla Firefox
192.168.0.16:8080/guacamole/#/

RECENT CONNECTIONS
bodhi
Windows 10
Ubuntu 18.10 Desktop

ALL CONNECTIONS
Filter
Ubuntu 18.10 Desktop — Currently in use by 1 user.
Windows 10 — Currently in use by 1 user.

Figure 4: In addition to a screenshot, you can view the number of users connected to each remote desktop session from the desktop.



Input method

None

No input method is used. Keyboard input is accepte connected, physical keyboard.

Text input

Allow typing of text, and emulate keyboard events based on the typed text. This is necessary for devices such as mobile phones that lack a physical keyboard.

On-screen keyboard

Display and accept input from the built-in Guacamole on-screen keyboard. The on-screen keyboard allows typing of key combinations that may otherwise be impossible (such as Ctrl-Alt-Del).

Mouse emulation mode

Determines how the remote mouse behaves with respect to touches.

Figure 5: You can use the onscreen keyboard to pass certain key combinations (such as Alt+Tab) to the remote desktop.

under the *System* section. You'll be taken to the *Remote* tab in the System Properties window, where you need to toggle the *Allow remote connections to this computer* option.

In Windows 10, the option for only allowing connections from PCs running Remote Desktop with Network Level Authentication is also enabled by default. Make sure you disable this option; otherwise, you'll not be able to connect to your Windows machine from Linux.

Furthermore, to access a Windows 10 machine, I had to toggle the *Ignore server certificate* option under the Authentication section in the Guacamole settings (Figure 3).

Remote Control

As mentioned earlier, all configured connections are listed in Guacamole's dashboard, along with icons of the most recently accessed ones (Figure 4).

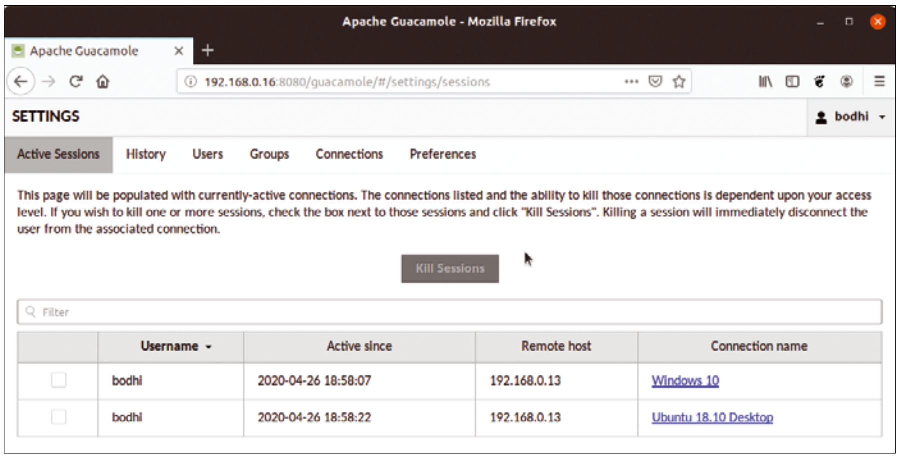
You can double-click a listed connection to launch a remote desktop session. Guacamole includes a hidden onscreen menu that you can bring up with the Ctrl + Alt + Shift key combination (Figure 5). If you are accessing Guacamole from a mobile device, you can bring up the menu by swiping from the left edge of the screen. The menu offers several features such as an on-screen keyboard, clipboard management, screen zoom control, and more.

Click your username in this menu to bring up more options. The *Disconnect* option will terminate the session, the *Home* option will take you to the Guacamole dashboard, and the *Settings* option will take you to the dashboard's Settings window.

The *Active Sessions* tab in the Settings window lists all sessions that you have exited without first disconnecting them. The sessions are listed in a sortable table along with various details such as the username, the duration of the active session, and more (Figure 6). You can terminate an active session by selecting its corresponding checkbox. After you've selected the sessions you wish to terminate, click the *Kill Sessions* button to disconnect them.

Sharing Is Caring

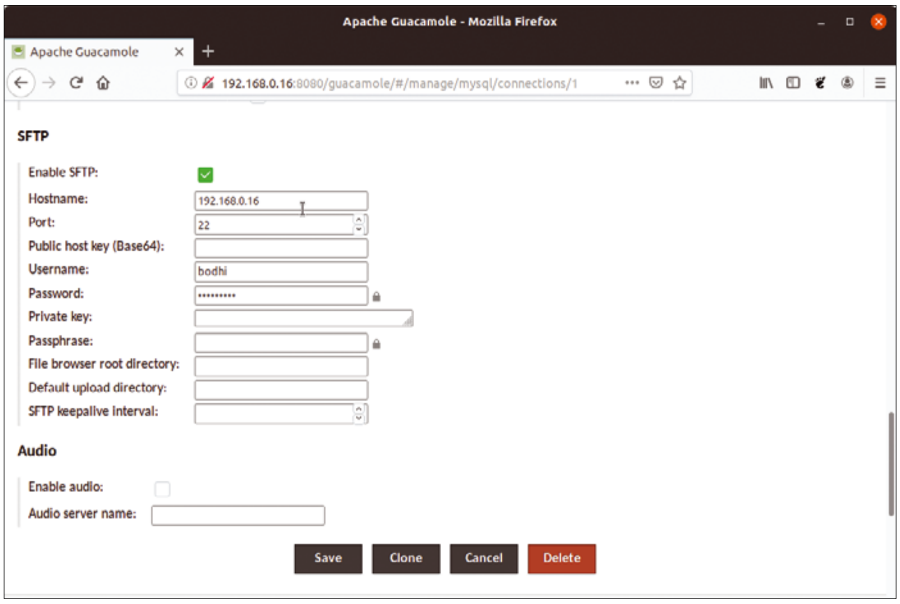
While Guacamole offers several options to tweak your remote desktop connections, one of the most essential is file sharing. You can easily transfer files back and forth between your local computer and the remote desktop. Cur-



This page will be populated with currently-active connections. The connections listed and the ability to kill those connections is dependent upon your access level. If you wish to kill one or more sessions, check the box next to those sessions and click "Kill Sessions". Killing a session will immediately disconnect the user from the associated connection.

	Username	Active since	Remote host	Connection name
☐	bodhi	2020-04-26 18:58:07	192.168.0.13	Windows 10
☐	bodhi	2020-04-26 18:58:22	192.168.0.13	Ubuntu 18.10 Desktop

Figure 6: While *Active Sessions* lists all ongoing sessions, switch to the *History* tab to view a list of recently used sessions.


Apache Guacamole - Mozilla Firefox
Apache Guacamole
192.168.0.16:8080/guacamole/#/manage/mysql/connections/1
Save
Clone
Cancel
Delete

SFTP

Enable SFTP:	✓
Hostname:	192.168.0.16
Port:	22
Public host key (Base64):	
Username:	bodhi
Password:	••••••••
Private key:	
Passphrase:	
File browser root directory:	
Default upload directory:	
SFTP keepalive interval:	

Audio

Enable audio:	
Audio server name:	

Figure 7: You can provide the optional SFTP Host key for Guacamole to verify the SFTP server's identity before transferring files.

rently, Guacamole supports file transfer for VNC, RDP, and SSH, using either the protocol's native file transfer support or SFTP.

To enable file transfer, bring up the Settings page for one of your remote connections. Scroll down to the *SFTP* section and toggle the *Enable SFTP* button (Figure 7). Enter the remote machine's hostname and *22* as the port since you are connecting using the SSH protocol. Finally, enter the authentication information and click the *Save* button to activate the settings.

You'll now be able to transfer files to the remote machine. You can either drag and drop them inside the browser window or use the file browser located in the Guacamole menu (Figure 8). Navigate the filesystem and then use the *Upload Files* button to select the file you wish to upload to the remote machine.

Once you have configured and accessed a remote connection, you can explore the various other parameters to tweak the connection per your requirements. Guacamole offers extensive options, and you can refer to its official documentation [6] for more information.

Info

[1] Apache Guacamole: *https://guacamole.apache.org*

[2] Installing Guacamole from source: *https://guacamole.apache.org/doc/gug/installing-guacamole.html*

[3] Installing Guacamole with Docker: *https://guacamole.apache.org/doc/gug/guacamole-docker.html*

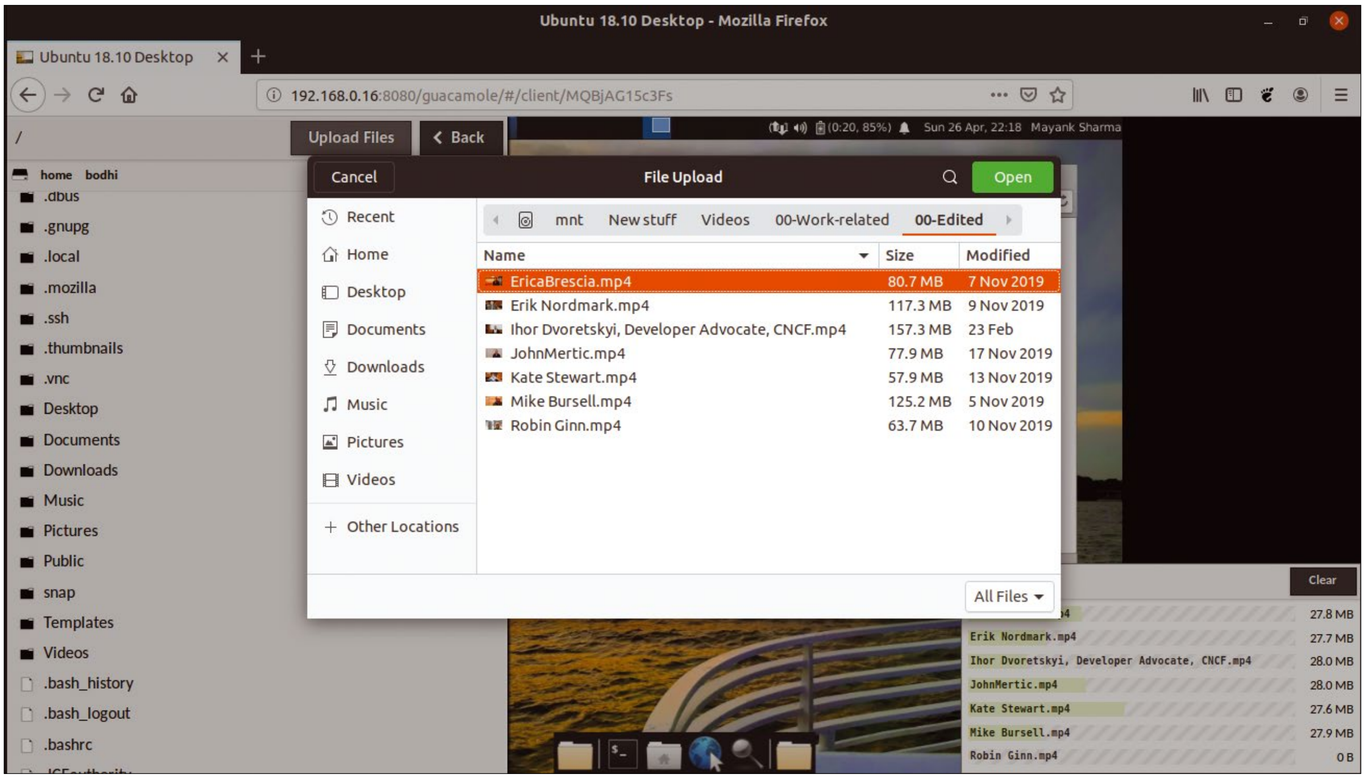
[4] Ubuntu script: *https://github.com/MysticRyuujin/guac-install*

[5] RHEL and CentOS script: *https://github.com/Zer0CoolX/guacamole-install-rhel*

[6] Guacamole documentation: *https://guacamole.apache.org/doc/gug/*

Author

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



/
home bodhi

.dbus
.gnupg
.local
.mozilla
.ssh
.thumbnails
.vnc
Desktop
Documents
Downloads
Music
Pictures
Public
snap
Templates
Videos
.bash_history
.bash_logout
.bashrc

Name	Size	Modified
EricaBrescia.mp4	80.7 MB	7 Nov 2019
Erik Nordmark.mp4	117.3 MB	9 Nov 2019
Ihor Dvoretskyi, Developer Advocate, CNCF.mp4	157.3 MB	23 Feb
JohnMertic.mp4	77.9 MB	17 Nov 2019
Kate Stewart.mp4	57.9 MB	13 Nov 2019
Mike Bursell.mp4	125.2 MB	5 Nov 2019
Robin Ginn.mp4	63.7 MB	10 Nov 2019

```
...4                                              27.8 MB
Erik Nordmark.mp4                                 27.7 MB
Ihor Dvoretskyi, Developer Advocate, CNCF.mp4     28.0 MB
JohnMertic.mp4                                    28.0 MB
Kate Stewart.mp4                                  27.6 MB
Mike Bursell.mp4                                  27.9 MB
Robin Ginn.mp4                                    0 B
```

Figure 8: A notification dialog tracks the status of all uploads, while the browser's download notification system tracks downloads.

IT Highlights at a Glance

HPC Up Close

- More Best Practices for HPC Containers
- 12th Annual Women in HPC Workshop
- US DOE Sponsors Quantum Internet Blueprint Workshop
- Serverless Computing With AWS Lambda

Further Reading

- High-Performance Python 4
- Thrashing the Data Cache for Fun and Profit
- Linux Writecache
- The New OpenShift Version 4
- ROCm 3.0 Arrives

Highlights

More Best Practices for HPC Containers
Best practices for building and running Docker and Singularity containers.

12th Annual Women in HPC Workshop
June 25 meetup at the ISC conference in Germany will offer mentoring for early-career women and information on inclusion for community leaders.

ors Quantum Internet Blueprint
up plans for a nationwide qubit network

puting With AWS Lambda
S Lambda serverless technology reduces
your infrastructure automatically.

to HIP
ry CUDA code new life with an open

nd AWS Use HPC to Build a Better

ADMIN HPC
HPC news, views, and technical articles delivered to your inbox every month.
Subscribe Now!
ADMIN Network & Security

ADMIN Update – Hottest Links

- VoIP and NAT
- Ubuntu 20.04 ZFS Includes apt Snapshots
- Samba 4.12 Released
- Loki
- Ansible in the Hybrid Cloud

Further Reading

- Systemd-homed is Coming to a Linux Distribution Near You
- Software RAID
- Open Source Software Dominates the Enterprise
- Container Management with Podman
- Nine Year Old Bug Found and Fixed in Sudo

Highlights

VoIP and NAT

We show you how to secure transparent IP address transitions through NAT firewalls and gateways for Voice over IP. (more)

Ubuntu 20.04 ZFS Includes apt Snapshots

The next release of Ubuntu Desktop will include a handy new ZFS feature. (more)

Samba 4.12 Released

The first stable release of the Samba 4.12 series improves the speed of encrypted file transfers. (more)

Loki

Grafana's Loki is a good replacement candidate for the Elasticsearch, Logstash, and Kibana combination in Kubernetes environments. (more)

Ansible in the Hybrid Cloud

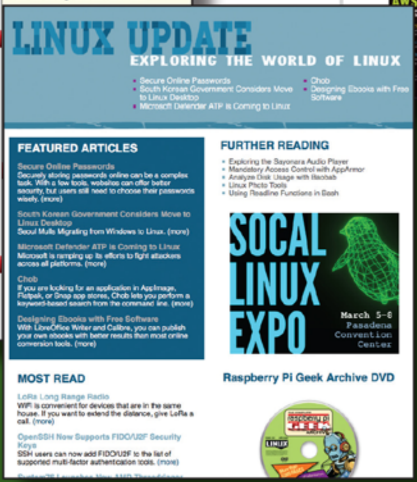
Extending your data center temporarily into the cloud during a customer rush might not be easy, but it can be done, thanks to Ansible's Playbooks and some AWS scripts. (more)

Most Read Articles

Linux Systems Vulnerable to Attack

Millions of Linux systems are vulnerable to attack, due to at-risk firmware (more)

Windows Subsystem for Linux



- Secure Online Passwords
- South Korean Government Considers Move to Linux Desktop
- Microsoft Defender ATP is Coming to Linux
- Chob
- Designing Ebooks with Free Software

FEATURED ARTICLES

Secure Online Passwords
Securely storing passwords online can be a complex task. With a few tools, websites can offer better security, but users still need to choose their passwords wisely. (more)

South Korean Government Considers Move to Linux Desktop
Seoul Mulls Migrating from Windows to Linux. (more)

Microsoft Defender ATP is Coming to Linux
Microsoft is ramping up its efforts to fight attackers across all platforms. (more)

Chob
If you are looking for an application in AppImage, Flatpak, or Snap app stores, Chob lets you perform a keyword-based search from the command line. (more)

Designing Ebooks with Free Software
With LibreOffice Writer and Calibre, you can publish your own ebooks with better results than most online conversion tools. (more)

FURTHER READING

- Exploring the Sayonara Audio Player
- Mandatory Access Control with AppArmor
- Analyze Disk Usage with Baobab
- Linux Photo Tools
- Using Readline Functions in Bash

MOST READ

LoRa Long Range Radio
WiFi is convenient for devices that are in the same house. If you want to extend the distance, give LoRa a call. (more)

OpenSSH Now Supports FIDO/U2F Security Keys
SSH users can now add FIDO/U2F to the list of supported multi-factor authentication tools. (more)

Raspberry Pi Geek Archive DVD

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

ADMIN HPC • ADMIN Update • Linux Update

and keep your finger on the pulse of the IT industry.

ADMIN and HPC: **bit.ly/HPC-ADMIN-Update**
Linux Update: **bit.ly/Linux-Update**

Photo by Parker Byrd on Unsplash

Build a VPN tunnel with WireGuard

Uncomplicated

A recent addition to the Linux kernel, WireGuard lets you build a VPN tunnel that relies on encryption to reduce potential security issues. *By Ferdinand Thommes and Christoph Langner*

As a result of the COVID-19 pandemic, many employees have exchanged the office for their home to accommodate social distancing guidelines. In addition to getting used to working from home, many telecommuters must also deal with security issues when contacting colleagues or accessing company servers. While large corporations may take care of these issues for their employees, self-employed telecommuters and small businesses need to find their own solution.

WireGuard [1], the modern virtual private network (VPN) tunnel software developed by security researcher Jason Donenfeld, offers an easy-to-implement solution that relies on encryption to secure the connection between two endpoints. WireGuard found its way into the Linux kernel 5.6 at the end of March at the same time WireGuard v1.0.0 was released. The VPN program is now available for all common operating systems such as Linux, macOS, Windows, Android, and iOS.

Competition

Before WireGuard conquered the market in 2015, IPsec and OpenVPN were the top two contenders under a free license. Compared to WireGuard, however, both IPsec and OpenVPN are more difficult to set up, which is why WireGuard was already in use before becoming a kernel module.

Linux Torvalds had hoped WireGuard would be merged to the kernel in 2018. In comparison to OpenVPN and IPSec, Torvalds has called WireGuard “a work of art” [2]. If you have followed Torvalds’ statements over the years, you know that he is generally very sparing with praise.

WireGuard gets by with only about 4,000 lines of source code. In comparison, OpenVPN together with the required OpenSSL weigh in at around 600,000 lines of code, while IPsec and StrongSwan use more than 400,000 lines. WireGuard offers far less attack potential than its competitors. The software also relies on modern algorithms: ChaCha20 [3] is used for encryption, while Curve25519 handles the key exchange [4].

Fast and Frugal

WireGuard shows its advantages over the established solutions in terms of speed and resource consumption. This manifests itself in far faster and more stable connections, especially when roaming. While OpenVPN often consumes 30 percent of battery power on Android, WireGuard keeps this in the lower single-digit range.

We tested WireGuard with Ubuntu 20.04 LTS, which comes with the backported module for WireGuard in kernel 5.4. Ubuntu users were already interested in WireGuard before its inclusion in the kernel, as evidenced by over 20,000 installations from the WireGuard PPA. There is also a backport for Debian 10 Buster.

Not Just for Linux

WireGuard can also be used with OpenBSD, FreeBSD, NetBSD, macOS, and Microsoft Windows (a stable version is imminent for Windows). For road warriors, there are apps for Android and iOS. You will want to use the original apps rather than third-party apps [5].

You can use WireGuard with modest hardware resources. In terms of the server, you don’t need anything faster than an older laptop, a single board computer like the Raspberry Pi, or a

Lead Image © Roman Sakhno, 123RF.com

rented V-Server on the web. In our test, we used a ThinkPad X220, a device that has been out of service for quite some time (see the box "DynDNS and Port Forwarding"). WireGuard supports constellations with two clients or with one server and multiple clients.

Tunneled

After completing the setup, the laptop, which acts as a server in our case, will take responsibility for transporting the network packets and will reside between the client and, for example, any websites it visits, accepting requests and returning responses. This connection is encrypted in both directions. Visited websites only see the server's IP address, not your own.

Setting up a VPN with WireGuard is easier than with its competitors (which sometimes require a demanding configuration that is easily beyond a beginner's capabilities). With the recent addition of WireGuard to the mainline kernel, its adoption is expected to continue to grow; over time, the configuration is likely to be simplified with additional tools.

Installing WireGuard

Unlike its competitors, WireGuard uses the same software on the server and the clients. After installing the *wireguard* package via the server's and the clients' package managers, start the process of generating private and public keys; this is comparable to the same procedure in SSH. You need to create a key pair for each device that will have access to the VPN. The two computers on either end of the WireGuard tunnel each need the public keys from the other end. WireGuard does not care whether the server is on the Internet or a local network.

If you are using Ubuntu 20.04, the best way to install WireGuard is to type the following at the command line

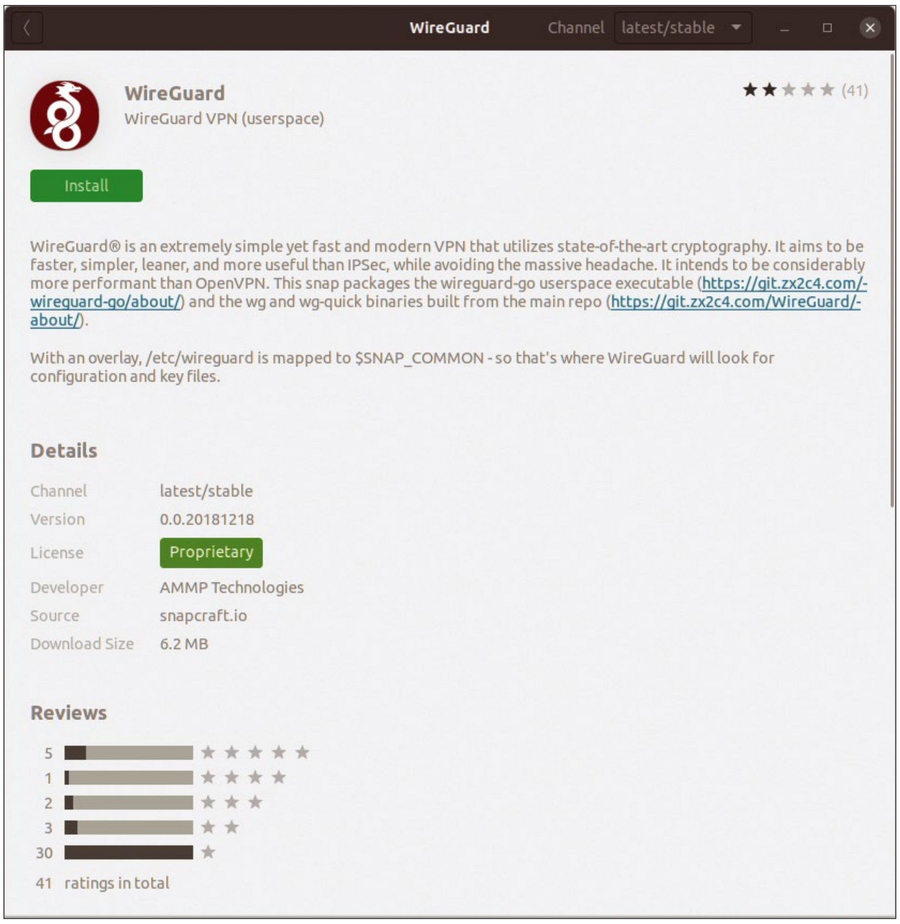
```
sudo apt install wireguard
```

rather than using the graphical package manager, which only gives you an outdated third-party snap package (Figure 1). Also make sure that the header files are installed to match the kernel.

After installing the package, you still need to enable IP forwarding on the designated WireGuard server. As root, open the `/etc/sysctl.conf` file in an editor and uncomment the lines `#net.ipv4.ip_forward=1` for IPv4 or `#net.ipv6.conf.all.`

DynDNS and Port Forwarding

A local VPN network on your own LAN only makes sense in very rare cases. The typical application scenario involves dialing into the company network or your home LAN from somewhere outside. In this scenario, you need a DynDNS address, provided by something like the free DynDNS Service [6]. You also need to forward the port used by WireGuard (in our example port 51820/UDP) from the WLAN router to the computer used as a server. Details of the required configuration are usually provided in your device's operating manual. In the case of a FRITZ!Box, call the device's administration interface by typing the FRITZ!Box URL in your browser and then open the wizard in *Internet* | *Shares* | *Port Shares* by clicking on *Add Device for Shares*, which helps you set up port forwarding.



WireGuard® is an extremely simple yet fast and modern VPN that utilizes state-of-the-art cryptography. It aims to be faster, simpler, leaner, and more useful than IPSec, while avoiding the massive headache. It intends to be considerably more performant than OpenVPN. This snap packages the wireguard-go userspace executable (https://git.zx2c4.com/-wireguard-go/about/) and the wg and wg-quick binaries built from the main repo (https://git.zx2c4.com/WireGuard/-about/).

With an overlay, /etc/wireguard is mapped to $SNAP_COMMON - so that's where WireGuard will look for configuration and key files.

Details

Channel	latest/stable
Version	0.0.20181218
License	Proprietary
Developer	AMMP Technologies
Source	snapcraft.io
Download Size	6.2 MB

Reviews

5	★★★★★
1	★★★★
2	★★★
3	★★
30	★

41 ratings in total

Figure 1: Ubuntu 20.04 LTS offers two different WireGuard packages. The Ubuntu Software Center gives you an outdated third-party snap. Instead, use the *wireguard* package, which you can install at the command line.

Listing 1: Enabling IP Forwarding

```
[...]
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
[...]
# Uncomment the next line to enable packet forwarding for IPv6
#  Enabling this option disables Stateless Address Autoconfiguration
#  based on Router Advertisements for this host
net.ipv6.conf.all.forwarding=1
[...]
```

Listing 2: Reloading WireGuard

```
### Install Wireguard
$ sudo apt update
$ sudo apt install wireguard
resolvconf
### Only on the Wireguard server:
$ sudo nano /etc/sysctl.conf
$ sudo sysctl -p
```

`forwarding=1` for IPv6 (Listing 1). Then reload the system configuration (Listing 2) by typing:

```
sudo sysctl -p
```

Key Services

Now create the required private and public keys on the server and clients (shown in Listing 3). Finally, check that the keys have been created with the `ls` command (Figure 2). It is best to copy both public keys into a text file and save them on a USB stick for later configuration.

Building an Interface

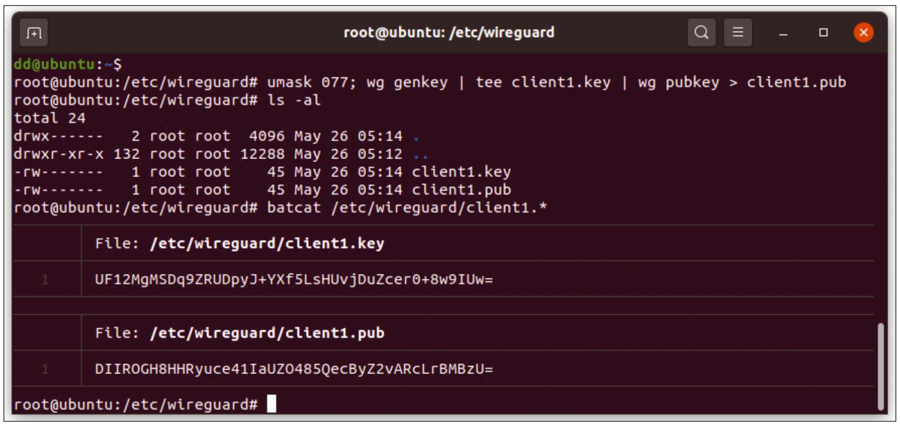
The next step is to create one virtual network interface per device for WireGuard. This is the equivalent of `eth0` or `wlan0`. However, since traffic is tunneled separately with WireGuard, the system requires one interface for routing. The default for the first interface in WireGuard is `wg0` (which we will use for simplicity's sake). If you prefer a different name, this is not a problem as long as you use it consistently.

The interface you created requires a specific IP range. In order to distinguish it from the 192.168.<x>.<y> common on LANs, use IPs from the private network address range 10.0.10.x [7] for `wg0`, which you then make available to the outside world via masquerading [8]. For the configuration, save Listing 4 (server) and Listing 5 (client) in the `/etc/wireguard/wg0.conf` file.

Be sure to enter the cryptographic keys from the key files. You will also want to tell the server to use the WLAN router as a DNS server using the `DNS = [...]` line. The WireGuard client, on the other hand, needs to use the `DNS = 10.0.10.1` option to query the WireGuard computer as its DNS server. The server configuration includes the same rules for starting up and shutting down the interface.

Listing 3: Creating Private and Public Keys

```
$ sudo -s
$ cd /etc/wireguard
### Generate key on server:
$ umask 077; wg genkey | tee <client1>.key | wg pubkey > <client1>.pub
### Generate key on client:
$ umask 077; wg genkey | tee <client2>.key | wg pubkey > <client2>.pub
### Check key on server:
$ ls -al
total 24
drwx------   2 root root  4096 Apr 30 19:49 .
drwxr-xr-x 131 root root 12288 Apr 30 19:47 ..
-rw-------   1 root root    45 Apr 30 19:49 client1.key
-rw-------   1 root root    45 Apr 30 19:49 client1.pub
$ cat /etc/wireguard/client1.key
YBwK1N1O7OwOEtWCFnxwF9aVBOGK5YUNxEtU1pyVuUs=
$ cat /etc/wireguard/client1.pub
LnEReQTHUY7FIMaAR6qVcCfk95ucPY6O/zb4OfdfYh4=
```



```
root@ubuntu: /etc/wireguard
dd@ubuntu:~$
root@ubuntu:/etc/wireguard# umask 077; wg genkey | tee client1.key | wg pubkey > client1.pub
root@ubuntu:/etc/wireguard# ls -al
total 24
drwx------   2 root root  4096 May 26 05:14 .
drwxr-xr-x 132 root root 12288 May 26 05:12 ..
-rw-------   1 root root    45 May 26 05:14 client1.key
-rw-------   1 root root    45 May 26 05:14 client1.pub
root@ubuntu:/etc/wireguard# batcat /etc/wireguard/client1.*
       File: /etc/wireguard/client1.key
       UF12MgMSDq9ZRUDpyJ+YXf5LsHUvjDuZcer0+8w9IUw=

       File: /etc/wireguard/client1.pub
       DIIROGH8HHRyuce41IaUZO485QecByZ2vARcLrBMBzU=
root@ubuntu:/etc/wireguard#
```

Figure 2: Creating the cryptographic keys is reminiscent of SSH. Repeat the commands on every computer that is involved.

Listing 4: Server Configuration

```
[Interface]
Address = 10.0.10.1/24,fd42:42:42::1/64
SaveConfig = true
PrivateKey = <Key from client1.key>
ListenPort = 51820
PostUp = iptables -A FORWARD -i wg0 -j ACCEPT; iptables -t nat -A POSTROUTING -j
           MASQUERADE; iptables -A FORWARD -o wg0 -j ACCEPT
PostUp = ip6tables -A FORWARD -i wg0 -j ACCEPT; ip6tables -t nat -A POSTROUTING -j
           MASQUERADE; ip6tables -A FORWARD -o wg0 -j ACCEPT
PostDown = iptables -F; iptables -t nat -F
PostDown = ip6tables -F; ip6tables -t nat -F
DNS = <IP Address of the WLAN Router>

[Peer]
PublicKey = <Key from client2.pub>
AllowedIPs = 10.0.10.2/32,fd42:42:42::/64
```

Listing 5: Client Configuration

```
[Interface]
Address = 10.0.10.2/32,fd42:42:42::2/64
PrivateKey = <Key from client2.key>
DNS = 10.0.10.1

[Peer]
PublicKey = <Key from client1.pub>
Endpoint = <beispiel.dyndns.com>:51820
AllowedIPs = 0.0.0.0/0,::/0
PersistentKeepalive = 20
```

Listing 6: Configuring Port Sharing with UFW

```
$ sudo ufw allow 22/tcp
$ sudo ufw allow <51820>/udp  ### Or the port chosen by the system
$ sudo ufw enable
$ sudo ufw status verbose
Status: Active
Logging: on (low)
Default: deny (incoming), allow (outbound), deny (sent)
New profiles: skip

To                          Action      From
--                          ------      ---
22/tcp                      ALLOW IN    Anywhere
51820/udp                   ALLOW IN    Anywhere
22/tcp (v6)                 ALLOW IN    Anywhere (v6)
51820/udp (v6)              ALLOW IN    Anywhere (v6)
```

WireGuard uses port 51820 as the default port. If you want to use a different port, adjust the `ListenPort = 51820` line accordingly. You then control the connection setup with the command:

```
sudo wg-quick up wg0
```

You can shut down the VPN again by typing:

```
sudo wg-quick down wg0
```

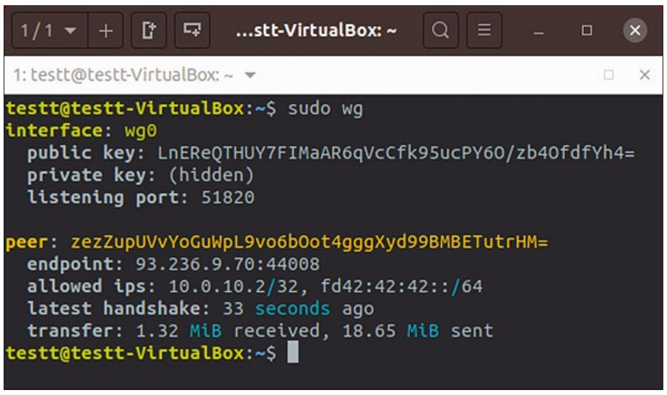
The `sudo wg` command tells WireGuard to display information on the current status (Figure 3). To activate `wg0` automatically at system startup time, type the following command on both computers:

```
sudo systemctl enable wg-quick@wg0
```

There is also the Gufw graphical interface. After installation, if necessary, allow port 22/TCP for SSH on both devices and open 51820/UDP or your chosen port (Figure 4).

Conclusions

If you have no errors in the configuration, you should be able to ping the other IP address. From our personal experience, we found that configuration errors can happen easily. However, WireGuard can be set up quite quickly with about an hour of concentrated work. You can then extend the configuration to include additional clients, such as Android smartphones, using the same principle. ■■■



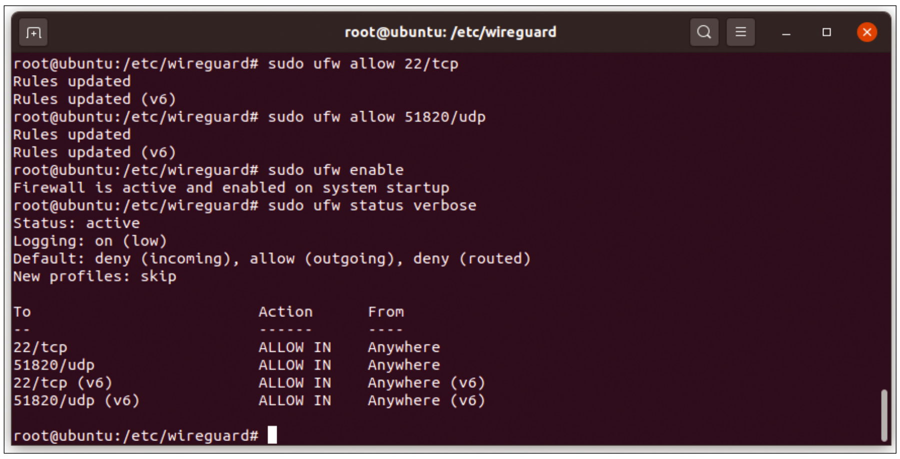
```
testt@testt-VirtualBox:~$ sudo wg
interface: wg0
  public key: LnEReQTHUY7FIMaAR6qVcCfk95ucPY6O/zb4OfdfYh4=
  private key: (hidden)
  listening port: 51820

peer: zezZupUVvYoGuWpL9vo6bOot4gggXyd99BMBETutrHM=
  endpoint: 93.236.9.70:44008
  allowed ips: 10.0.10.2/32, fd42:42:42::/64
  latest handshake: 33 seconds ago
  transfer: 1.32 MiB received, 18.65 MiB sent
testt@testt-VirtualBox:~$
```

Figure 3: The `sudo wg` command delivers information and statistics on the status of your WireGuard VPN.

UFW

The easiest way to configure the port sharing settings is via the Uncomplicated Firewall (UFW) iptables front end (Listing 6), which is preinstalled on Ubuntu and included in most distributions' package sources.



```
root@ubuntu:/etc/wireguard# sudo ufw allow 22/tcp
Rules updated
Rules updated (v6)
root@ubuntu:/etc/wireguard# sudo ufw allow 51820/udp
Rules updated
Rules updated (v6)
root@ubuntu:/etc/wireguard# sudo ufw enable
Firewall is active and enabled on system startup
root@ubuntu:/etc/wireguard# sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), deny (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW IN    Anywhere
51820/udp                  ALLOW IN    Anywhere
22/tcp (v6)                ALLOW IN    Anywhere (v6)
51820/udp (v6)             ALLOW IN    Anywhere (v6)

root@ubuntu:/etc/wireguard#
```

Figure 4: Using UFW, you can allow access via SSH and open up the port for WireGuard. Setting up the iptables firewall is not absolutely necessary, but it does give the system additional security.

Info

[1] WireGuard: *https://www.wireguard.com/*

[2] "Linux Torvalds Is Hoping WireGuard Will Be Merged Sooner Rather Than Later" by Michael Larabel, August 3, 2018, *https://www.phoronix.com/scan.php?page=news_item&px=Linus-Likes-WireGuard*

[3] ChaCha20: *https://cr.yp.to/chacha.html*

[4] Curve25519: *https://cr.yp.to/ecdh.html*

[5] Installation: *https://www.wireguard.com/install*

[6] DynDNS Service: *https://dyndnss.net/eng/*

[7] Private IP addresses: *https://en.wikipedia.org/wiki/Private_network*

[8] Masquerading: *https://www.tldp.org/HOWTO/IP-Masquerade-HOWTO/ipmasq-background2.1.html*

6 issues per year!

GET IT FAST with a digital subscription!

shop.linuxnewmedia.com

REAL SOLUTIONS *for* REAL NETWORKS

ADMIN is 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.

Improve your admin skills with practical articles on:

- **Security**
- **Cloud computing**
- **DevOps**
- **HPC**
- **Storage and more!**

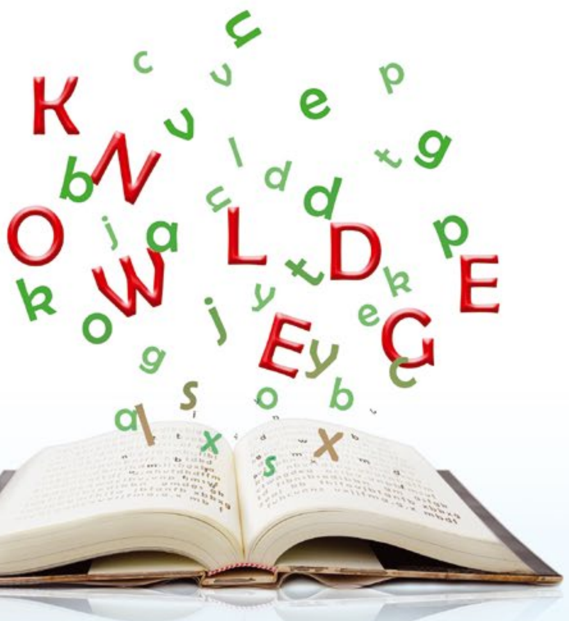
SUBSCRIBE NOW
SAVE 30%

ADMIN
Network & Security

shop.linuxnewmedia.com

Converting text to speech in LibreOffice and OpenOffice

Conversationalist



Visually impaired users often find working with text and tables in office suites difficult. Pico TTS, a text-to-speech synthesizer, and the Read Text extension for LibreOffice and OpenOffice provide a solution. *By Erik Bärwaldt*

Without special aids, people with vision impairments can find using a computer difficult. While icons can be sufficiently enlarged to make them readable, word processors and spreadsheets are a major hurdle. The fonts in the documents are almost always too small to be read and deciphering text is also difficult due to the smooth transitions in serif or cursive fonts.

In this case, programs that read the text out loud can help. These programs convert letters into linguistically adapted phonemes and then play them back via the computer's sound system with an installed voice.

On Linux, there are various screen readers for this purpose that are based on text-to-speech programs such as eSpeak or Festival. Solutions based on the popular eSpeak have the disadvantage of a synthetic computer voice: It is quite difficult to understand due to pronunciation that is typically very nasal and partly uses the wrong intonation.

Since software packages also need to be localized for the phonemes' linguistic modification, some packages are only usable for certain languages. For example, software only localized in German would render texts that are incomprehensible to Spanish-speaking listeners.

One of the most mature applications for text-to-speech synthesis is Pico TTS [1], a text-to-speech synthesizer originally developed by SVox and used by Google in Android. The command-line program can convert texts in several languages into `.wav` files. With the help of an extension for LibreOffice and OpenOffice, Pico can read out text or tables without requiring the user to enter clumsy command sequences in the terminal.

Listing 1: Installing Pico

```
$ sudo apt-get install libttspico0 libttspico-utils libttspico-data
```

Speech Synthesizer

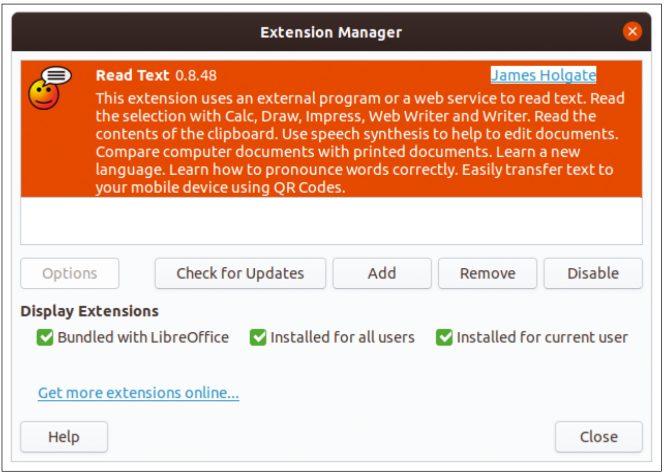
On Debian, Ubuntu, and their derivatives, you install Pico with the command shown in Listing 1. For other distributions, you can find packages whose names usually start with the string *svox-pico*. These packages require several *libttspico0* or *lib64ttspico0* packages as dependencies. If Pico is not currently available for your particular distribution, packages from other derivatives of the same distribution base can often be used.

To integrate the synthesizer into LibreOffice or OpenOffice, you need the Read Text extension for your choice of office suite. You can find the Read Text extension, suitable for almost all LibreOffice 6.x versions, on the LibreOffice Extensions page [2]. However, in testing, using Read Text 0.8.48 with the brand new LibreOffice 6.4.1 failed. There were no problems with the older 6.0.3 release.

To integrate the downloaded extension directly into LibreOffice, first open the *Extras* | *Extension Manager* dialog. In the dialog that opens (Figure 1), press the *Add* button and select the `read_text.2020.03.07.oxt` extension in the file browser that appears. LibreOffice will then automatically include the extension. Upon closing the window, it will prompt you to restart the office package to activate the extension.

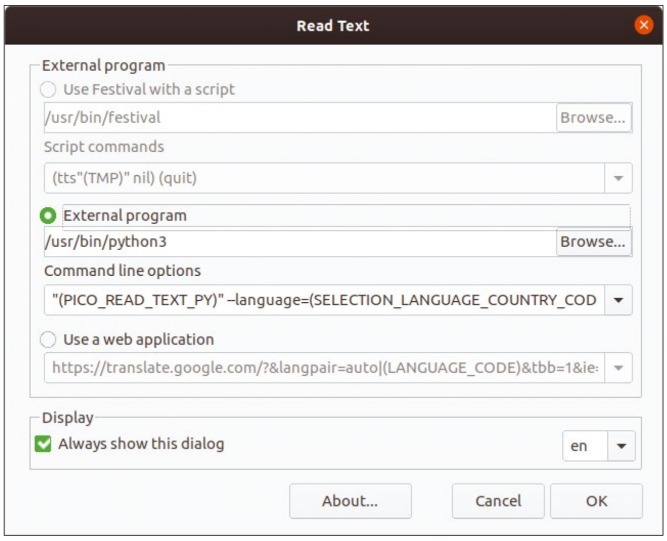
Reading Lesson

After the restart, you can continue working with LibreOffice as before. To have a section of text read aloud, select the *Read Selection* option in LibreOffice Writer via the *Tools* | *Add-Ons* menu, which opens the *Read Text* settings dialog (Figure 2).



Read Text 0.8.48 James Holgate
This extension uses an external program or a web service to read text. Read the selection with Calc, Draw, Impress, Web Writer and Writer. Read the contents of the clipboard. Use speech synthesis to help to edit documents. Compare computer documents with printed documents. Learn a new language. Learn how to pronounce words correctly. Easily transfer text to your mobile device using QR Codes.

Figure 1: The extension is installed via the Extension Manager.


Read Text

External program

Use Festival with a script

/usr/bin/festival — Browse...

Script commands

(tts"(TMP)" nil) (quit)

[✓ External program]

/usr/bin/python3 — Browse...

Command line options

"(PICO_READ_TEXT_PY)" –language=(SELECTION_LANGUAGE_COUNTRY_COD

Use a web application

https://translate.google.com/?&langpair=auto|(LANGUAGE_CODE)&tbb=1&ie=

Display

[✓ Always show this dialog] — en

Figure 2: You can enable voice output via an easy-to-use dialog.

First, define the speech synthesizer to be used. Festival always appears first, but it is grayed out if the software is not installed. The *External program* radio button is enabled if you are using Pico and eSpeak and lists the path to Python in an input field. Older distributions show `/usr/bin/python` as the path, while newer distributions use `/usr/bin/python3` as shown in Figure 2. You do not need to change this setting, because your system automatically determines the paths and the Python version used.

In the *Command line options* selection field, first click on the drop-down menu icon to the right and then select a speech synthesizer with the matching options. Even if only one synthesizer is installed, it is available for selection with several settings parameters. For some distributions, you can use the parameters in this field to define, for example, whether the system should save the speech output to an audio file. In some cases, various parameterized language settings can also be selected.

If the speech output is not satisfactory during a test, you can also call Google Translate in the web browser via the *Use a web application* radio button, which defaults to Google Translate. The selected text is then automatically sent to Google Translate. All you have to do is click on the speaker icon in the browser to start Google's speech synthesizer.

To get started with voice output, press the *OK* button in the bottom right-hand corner of the dialog. This closes the window, and the speech synthesizer starts to output the selected text.

OpenOffice

OpenOffice also supports speech output functions. The project even provides a Read Text extension for OpenOffice 2.x versions [3], which are more than 10 years old. With Pico, even older versions of OpenOffice can achieve usable speech output. However, the current 0.8.49 version of the Read Text extension is only compatible with the newer OpenOffice 4.x versions.

You install the extension similar to LibreOffice. You call the Extension Manager in the start window via the *Tools* menu. Although it offers far fewer setting options than its LibreOffice counterpart, it has the same basic functions. After setting up the extension, call up the settings dialog via the *Tools* | *Add-Ons* menu. The dialogs are the same as those in LibreOffice.

Unlike LibreOffice, OpenOffice lets you experiment with different language variants of the Pico synthesizer with older Read Text versions. To read English text aloud, for example, you can use one of the English language options offered by Read Text. The extension supports several English, French, Italian, German, and Spanish pronunciation options (Figure 3).

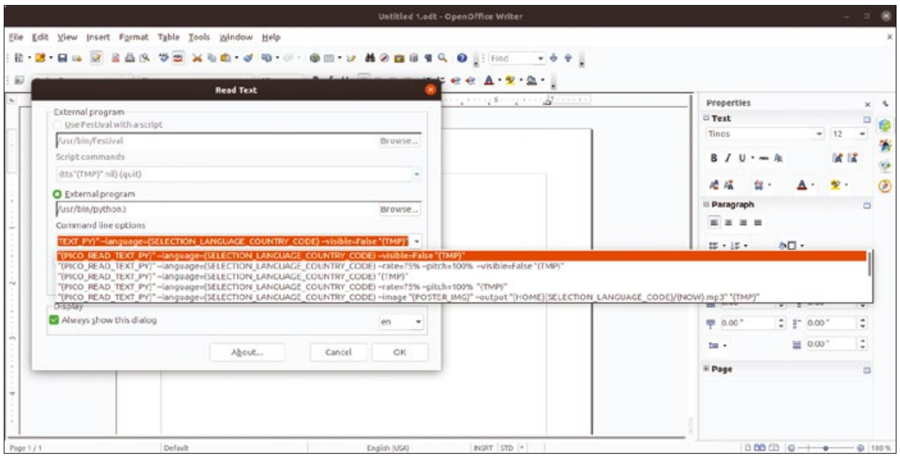
Google Translate can also be used as an alternative in OpenOffice by enabling the *Use a web application* option. If needed, you can also call another service provider to read the text by entering a new URL in the address line.

Clipboard

In both office suites, you can also play back texts from the clipboard. To have the clipboard read aloud, go to the *Tools* | *Add-Ons* menu. After selecting some text and copying it to the clipboard, select the *Read clipboard* option. The speech synthesizer then starts playing back the text immediately without displaying an additional dialog.

Conclusions

Using Pico and the Read Text extension lets you have text from your office suite read in the language of your choice. Both LibreOffice and OpenOffice also offer the Read Text extension for older versions of their office suites, so you don't have to use the latest version. Together, Pico and the Read Text extension make a valuable contribution to removing barriers for Linux users with special needs. ■■■


Untitled 1.odt - OpenOffice Writer

Read Text

External program

Use Festival with a script

/usr/bin/festival — Browse...

Script commands

(tts"(TMP)" nil) (quit)

[✓ External program]

/usr/bin/python3 — Browse...

Command line options

```
TEXT_PY)" –language=(SELECTION_LANGUAGE_COUNTRY_CODE) –visible=False "(TMP)"
"(PICO_READ_TEXT_PY)" –language=(SELECTION_LANGUAGE_COUNTRY_CODE) –visible=False "(TMP)"
"(PICO_READ_TEXT_PY)" –language=(SELECTION_LANGUAGE_COUNTRY_CODE) –rate=75% –pitch=100% –visible=False "(TMP)"
"(PICO_READ_TEXT_PY)" –language=(SELECTION_LANGUAGE_COUNTRY_CODE) "(TMP)"
"(PICO_READ_TEXT_PY)" –language=(SELECTION_LANGUAGE_COUNTRY_CODE) –rate=75% –pitch=100% "(TMP)"
"(PICO_READ_TEXT_PY)" –language=(SELECTION_LANGUAGE_COUNTRY_CODE) –image "(POSTER_IMG)" –output "(HOME)(SELECTION_LANGUAGE_CODE)/(NOW).mp3" "(TMP)"
```

Display

[✓ Always show this dialog] — en

Figure 3: The speech output settings in OpenOffice 3.2.1 are less extensive than LibreOffice, but they are just as useful.

Info

[1] Pico: *https://github.com/naggety/picotts*

[2] LibreOffice Read Text extension: *https://extensions.libreoffice.org/extensions/read-text*

[3] OpenOffice Read Text extension: *https://extensions.openoffice.org/en/project/read-text*

The Basics of Version Control

Git Ready

If you develop open source software, you need to know how to use Git. Information on Git fills books, but this basic overview will get you started with a vital open source development tool.

By Bruce Byfield

If Linus Torvalds were not already famous for Linux, he would be almost as famous for Git [1]. Within a few years of its first release in 2005, Git had become the dominant version control system in free software, replacing CVS and eclipsing rivals like Mercurial. Part of Git's popularity is no doubt due to Torvalds' own reputation, but a large part is also its highly organized structure: Every copy of Git is a complete repository with history and version-tracking tools that operates without a network or centralized server, giving it unmatched flexibility.

Author

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

Torvalds began Git after BitKeeper, the proprietary version control system that the Linux kernel project had begun using in 2002, withdrew permission to use it freely, claiming that Andrew Tridgell had reverse engineered another version control system from it [2]. Because version control is a necessity for development, Git was functional within a few weeks. Torvalds jokes that "git" is English slang for an unpleasant person, and the command is therefore named for himself, just as Linux is [3]. More obviously, "git" is the pronunciation of "get" in some American dialects. The truth is, after a few months, Torvalds passed maintenance of the project over to Junio Hamano, and Git has continued to grow in complexity ever since.

Git's basic purpose is to create an environment for developing files – usually code, but sometimes text as well. Entire books have been written about Git, but here is an overview to get you oriented. For all the detail, the basics are actually simple and become more so with practice.

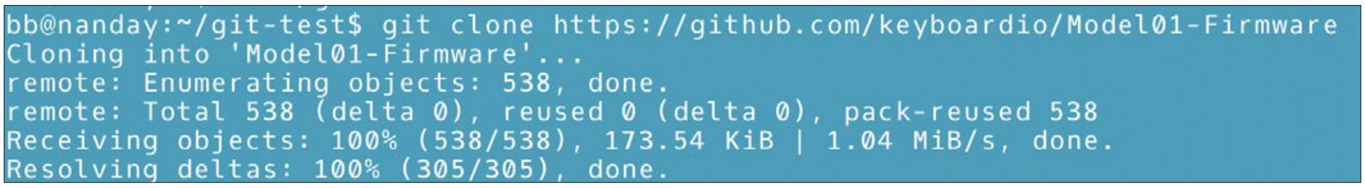
Setting Up a Repository

Git's use in the kernel guarantees that it is available in every major distribution. The easiest way to set up a local repository is to clone a repository online using the command:

```
git clone URL
```

This command uses the HTTPS protocol to create a directory with the same name as the last directory in the URL (Figure 1). You can use another protocol, including `git`, or specify a new name for the directory at the end of the command. This method has the advantage of reproducing a complex structure with a single command. In cases like Arduino firmware, this advantage is essential, because to flash your hardware generally requires a set

Lead Image © Raul Franganillo Fernandez, 123RF.com



```
bb@nanday:~/git-test$ git clone https://github.com/keyboardio/Model01-Firmware
Cloning into 'Model01-Firmware'...
remote: Enumerating objects: 538, done.
remote: Total 538 (delta 0), reused 0 (delta 0), pack-reused 538
Receiving objects: 100% (538/538), 173.54 KiB | 1.04 MiB/s, done.
Resolving deltas: 100% (305/305), done.
```

Figure 1: **The `clone` subcommand is a quick way to set up and configure a Git repository.**

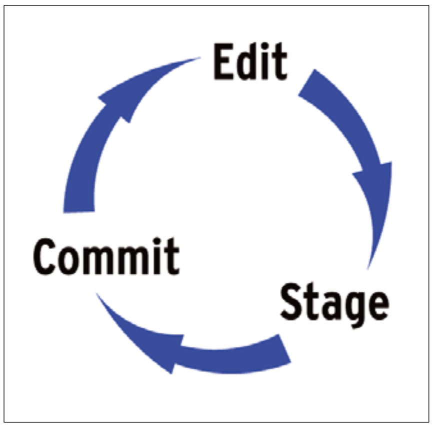


Figure 2: **The cyclical life cycle of a tracked file.**

structure for directories or files. To keep the local repository in sync with the original, type:

```
git pull URL
```

Updated files will be downloaded and merged, and new files will be downloaded.

To start a new repository, create a directory and change to the directory. Typing

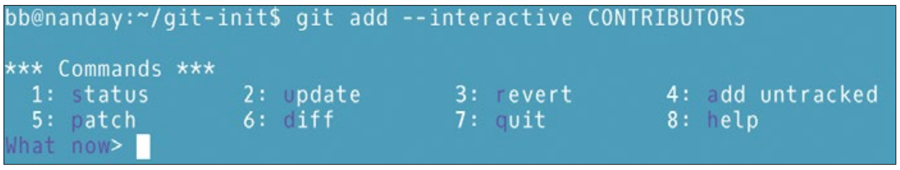
```
git init
```

creates a skeletal repository that is ready to receive files. If the directory already has files, they are drawn into the repository. You will also want to configure the new repository with:

```
git config --global user.name "NAME"
git config --global user.email "eMAIL"
```

Other configuration options are also available, including the setting of display colors and the default text editor [4]. When your local repository is set up, `git status` will give you a brief message: *Initialized empty Git repository in DIRECTORY*.

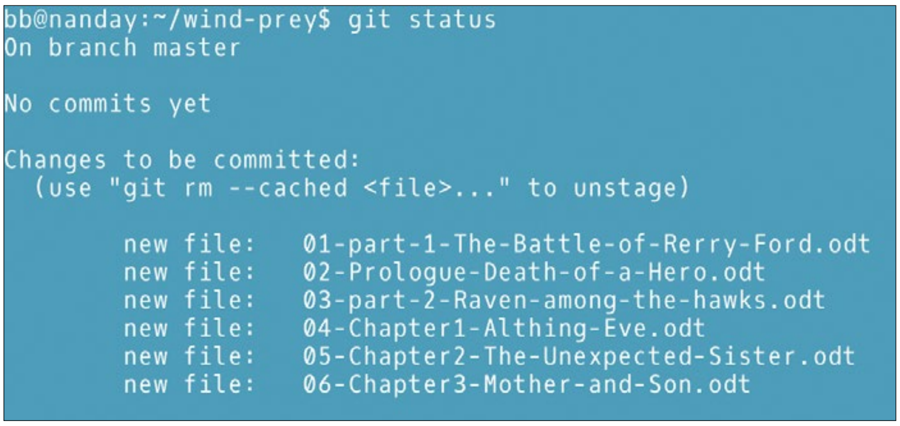
A third alternative is to use an online host like GitHub or GitLab and follow directions after creating an account. This is the most popular choice when you are working with others, especially if you do not have the hardware to host a project yourself or would prefer not to worry about your privacy and security.



```
bb@nanday:~/git-init$ git add --interactive CONTRIBUTORS

*** Commands ***
  1: status       2: update       3: revert       4: add untracked
  5: patch        6: diff         7: quit         8: help
What now> 
```

Figure 3: **`add`'s `--interactive` option lets you choose your next command from a table.**



```
bb@nanday:~/wind-prey$ git status
On branch master

No commits yet

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

        new file:   01-part-1-The-Battle-of-Rerry-Ford.odt
        new file:   02-Prologue-Death-of-a-Hero.odt
        new file:   03-part-2-Raven-among-the-hawks.odt
        new file:   04-Chapter1-Althing-Eve.odt
        new file:   05-Chapter2-The-Unexpected-Sister.odt
        new file:   06-Chapter3-Mother-and-Son.odt
```

Figure 4: **The `status` subcommand shows the complete state of the repository.**

Staging Files

A repository can have two types of files: tracked files, which have been added to Git, and untracked files, which have not. Untracked files may be works in progress that are not ready to be added to Git and may be edited freely. You can view the state of all files using `git status`, but Git's chief purpose is to manage tracked files. Tracked files go through a cyclical life cycle until they are removed as shown in Figure 2.

Adding a file to Git is often called staging. Staging simply means that Git is aware of a file and ready to work with it. The basic structure is:

```
git add FILE1 FILE2
```

A more precise commit can be made by using options, such as `--interactive` (`-i`) or `--patch` (`-p`) [5]. If you use `--interactive`, you receive a table for fast selection of your next command (Figure 3).

In contrast to adding a file, removing one requires some caution. The command:

```
git rm FILE
```

will remove a file from your local hard drive, as well as from Git. Often, though, you may want to stop Git from being aware of the file, but not to delete it from your system, in which case the command you need is:

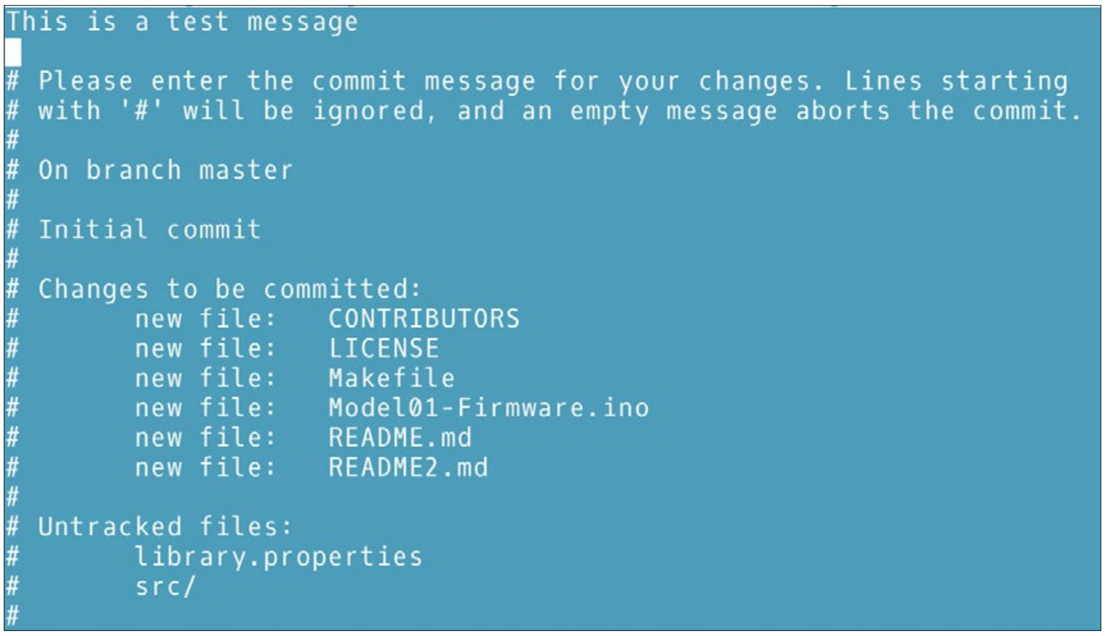
```
git rm --cached FILE
```

Making a Commit

Once a file is added, edited, or removed, you need to commit it. A commit is a snapshot of the repository, typically made each time after at least one file has been changed. The command `git status` lists all staged but uncommitted files (Figure 4). You can commit groups of files with the same command:

```
git commit -m "MESSAGE"
```

You will probably want to add to the command the option `--author="AUTHOR"` so that you take responsibility for your changes and receive credit for them as well. If you do not use the `-m` option, the editor in Git's environment opens so that you can write a detailed description (Figure 5). A repository's initial state is referred to as the `HEAD`, while a commit generally begins a new development branch of the code. For example, in the Linux kernel, each version number has its own branch.



```
This is a test message

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch master
#
# Initial commit
#
# Changes to be committed:
#       new file:   CONTRIBUTORS
#       new file:   LICENSE
#       new file:   Makefile
#       new file:   Model01-Firmware.ino
#       new file:   README.md
#       new file:   README2.md
#
# Untracked files:
#       library.properties
#       src/
#
```

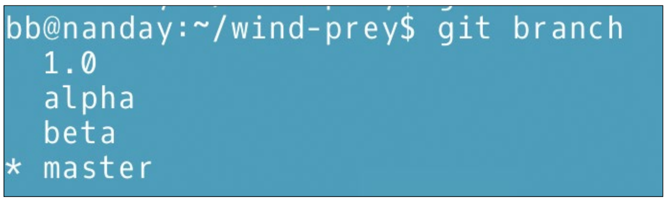
Figure 5: Before a commit is entered, Git summarizes the repository's state and gives you the chance to describe the changes you have made.

A commit is needed each time a file is added, edited, or removed. As commits are made, they are listed when `git status` is run. Changes from one commit to another can be read using `git diff` to see what has been done to the complete repository; you can also use `git diff FILE`.

Working with Branches

An important version control concept is branches, which are different versions of the same sets of files. Branches may be organized according to the stage of development – for instance, alpha, beta, and release – or by contributors or any other useful criteria. Especially in large projects with lots of files and users, it is more useful to work with branches as collections of files rather than with individual files, because changes to one file often affect others, and you want to be sure that your changes affect only those you intend to change.

To view a list of branches in numerical and alphabetical order, enter `git branch` (Figure 6).The output will include a branch called `master`, which is the main branch created with the repository, with an asterisk beside the branch that is currently active. To create a new branch, enter:



```
bb@nanday:~/wind-prey$ git branch
  1.0
  alpha
  beta
* master
```

Figure 6: The essential element of version control is branches: related versions of the same file.

```
git branch BRANCH-NAME
```

To switch to another branch, enter

```
git checkout BRANCH-NAME
```

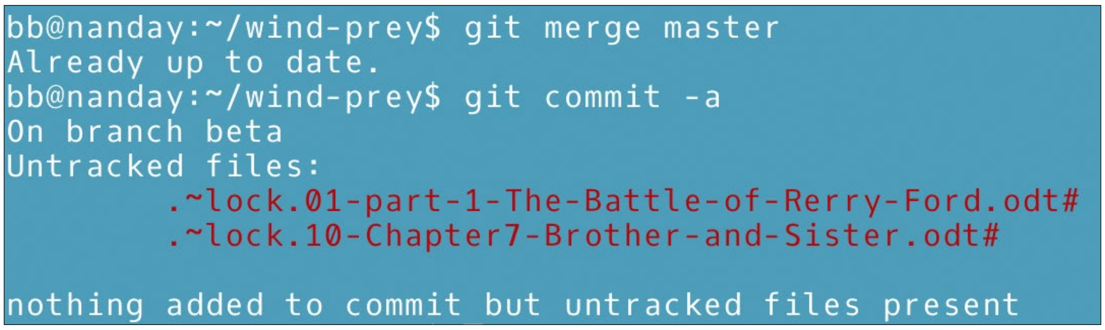
All your changes to files will affect only the versions of the files on the current branch. Each branch can have different versions of files with the same name. You can merge the file versions of the branch you are currently in with the file versions in another branch with:

```
git merge BRANCH-NAME
```

If there are any conflicts between the two branches, you can use `git diff` to view and reconcile them. When all conflicts are resolved, use

```
git commit -a
```

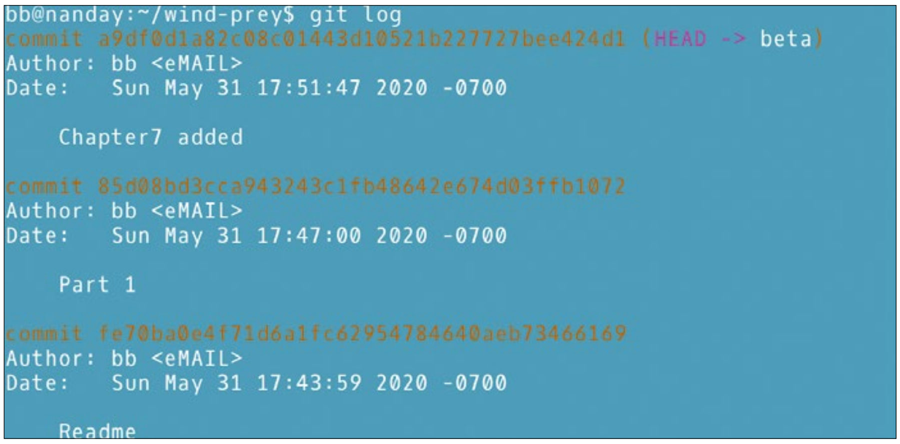
to commit the changes (Figure 7). The command `gitk` will show a graphical depiction of the merge. Any branches no longer needed can be removed with:



```
bb@nanday:~/wind-prey$ git merge master
Already up to date.
bb@nanday:~/wind-prey$ git commit -a
On branch beta
Untracked files:
        .~lock.01-part-1-The-Battle-of-Rerry-Ford.odt#
        .~lock.10-Chapter7-Brother-and-Sister.odt#

nothing added to commit but untracked files present
```

Figure 7: After working on one branch, you can merge it with another, either to transfer changes or (as here) to check your work.



```
bb@nanday:~/wind-prey$ git log
commit a9df0d1a82c08c01443d10521b227727bee424d1 (HEAD -> beta)
Author: bb <eMAIL>
Date:   Sun May 31 17:51:47 2020 -0700

    Chapter7 added

commit 85d08bd3cca943243c1fb48642e674d03ffb1072
Author: bb <eMAIL>
Date:   Sun May 31 17:47:00 2020 -0700

    Part 1

commit fe70ba0e4f71d6a1fc62954784640aeb73466169
Author: bb <eMAIL>
Date:   Sun May 31 17:43:59 2020 -0700

    Readme
```

Figure 8: Git's log can be a useful guide as you work.

```
git branch -d BRANCH-NAME
```

If you are working with branches of repositories that are remote from each other, use

```
git pull URL
```

to copy files from one branch to another. If conflicts result, branches are not merged until problems are resolved locally. The keeper of one repository can view useful information from the other one with:

```
git fetch URL master
git log -p HEAD..FETCH_HEAD
```

`FETCH_HEAD` refers to the remote head.

Using Log Information

Resolving conflicts between branches is easier with information from Git's log. The command `git log` shows a complete history of changes, while the addition of the `-p` option displays diffs, and the combo of `--stat` and `--summary` gives a convenient overview (Figure 8).

The log in Figure 8 identifies a commit with a 40 character name consisting of letters and numbers. Mercifully, you do not need to use the entire name, but can often use only enough of the name to identify the commit as unique or use `git tag` to give the commit a human-friendly name. In any of these ways, you can then use

```
git show COMMIT
```

to zero in on its details. In addition, `git grep` offers a strong search command, while

```
git log INCLUDED-BRANCH..EXCLUDED-BRANCH
```

restricts the branches to include in a search.

Next Steps

This overview should be enough to get you started with Git. However, there are countless additional details. Once you are comfortable with the basics, start looking up the options for Git's subcommands and gradually expand your range of choices.

Take advantage, too, of Git's exhaustive man pages. Perhaps because version control is essential to development, Git is one of the most documented commands in all of Linux. Besides the man page for the command itself, the two-part gittutorial(7) is a lengthy lesson in itself. It is supported by gitcvs-migration(7), gitcore-tutorial(7), gitglossary(7), git-help(1), gitworkflows(7), giteveryday(7), and The Git User's Manual[1] – if that is not enough, each of dozens of subcommands also has its separate man page.

One potentially puzzling aspect of this documentation is that it often classifies commands into porcelain (high-level commands about managing repositories like `git branch` or `git checkout`) and plumbing (low-level commands about working with files and branches like `git apply` and `git merge FILES`). This distinction does not affect Git's various tasks, but it can puzzle newcomers.

Aside from the porcelain/plumbing distinction, Git is complex, but its complexity is due to its size rather than structure. Once you understand the different types of tasks it is designed for, Git is not that hard to learn. And when you grasp it, you will understand one of the vital tools – if not rituals – of open source development. ■■■

Info

[1] Git: *https://git-scm.com/*

[2] Git history: *https://www.pcworld.idg.com.au/article/129776/after_controversy_torvalds_begins_work_git_/*

[3] Naming of Git: *https://www.youtube.com/watch?v=4XpnKHJAok8*

[4] Configuring a repository: *https://git-scm.com/docs/git-config*

[5] Git add: *https://git-scm.com/docs/git-add*

MakerSpace

Testing the Adafruit PyPortal touchscreen

Display Deluxe

Unlike other displays for the Raspberry Pi, Adafruit's PyPortal touchscreen provides an autonomous environment, including a microprocessor, sound output, and a WiFi connection. By Bernhard Bablok

The Raspberry Pi has a hard time with small displays because support from Raspbian is surprisingly poor. Creating and implementing suitable interfaces is difficult with a dearth of off-the-shelf programs for controlling small displays. I'm quietly confident that the PyPortal intelligent touchscreen by Adafruit is a better solution.

PyPortal (Figure 1) is a small 3.2-inch networkable resistive touch display with an integrated microprocessor [1]. At $55 (EUR59), it is not exactly cheap, but considering its components, the price seems reasonable. A correspondingly sized 3.2-inch display plus a Pi Zero W with an SD card, at about the same price, will serve as a comparison with the PyPortal configuration.

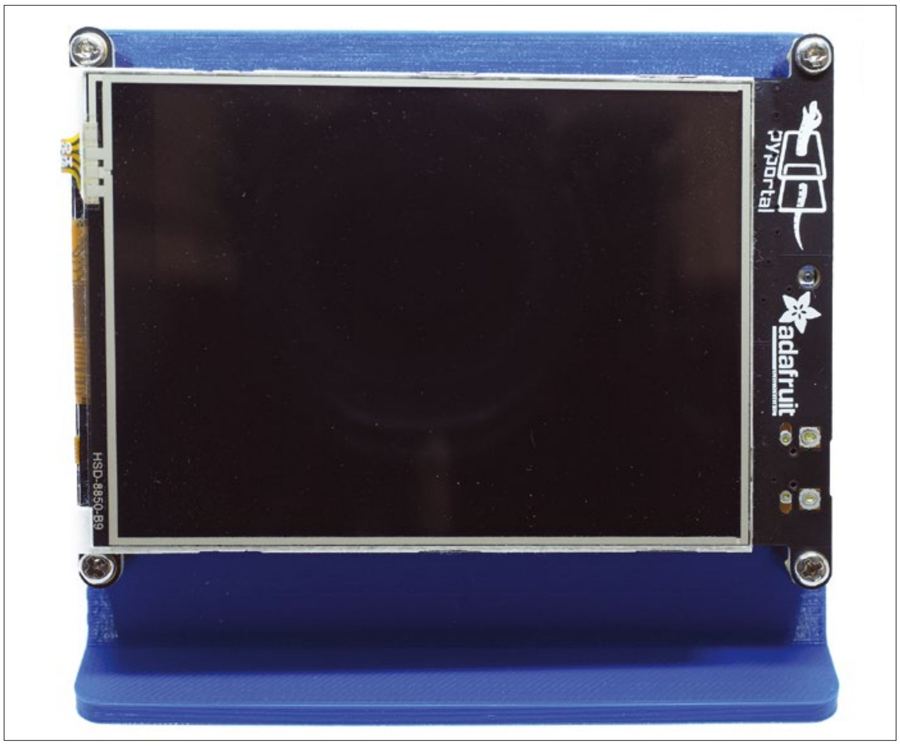


Figure 1: The front of the intelligent PyPortal display. The resistive touchscreen requires a certain amount of pressure before it reacts.

PyPortal in Detail

The screen takes up almost all of the real estate on the front, with a mounting frame and a brightness sensor on the right side. Adafruit has cut costs on the display itself. The resolution of 320x240 pixels is more suitable for 2.8-inch devices. Resistive touch technology is also a way of saving cash, but it does mean you need a stable mounting – just touching the screen is not enough, you actually have to press it.

The PyPortal is now available in two other sizes. The PyPortal Pynt for $45 reduces the screen diameter to 2.4 inches with the same number of pixels; the 3.5-inch PyPortal Titano offers a resolution of 320x480 pixels, plus a USB C power connection for $60.

Lead Image © donatos1205, 123RF.com

The back of the display (Figure 2) accommodates, among other things, an ATSAMD51J20 CPU by Atmel. It is supported by the far larger ESP32 WiFi coprocessor by Espressif. These popular modules provide WiFi at a low cost. The ESP32 takes on the computationally intensive TLS/SSL encryption protocol, thus reducing the burden on the Atmel CPU. Its Cortex-M4 processor runs at 120MHz and has 1MB of flash memory for program code and 256KB of RAM. Additionally, 8MB of flash memory is available for other resources, such as images or sounds.

In addition to the previously mentioned light sensor, the PyPortal's features include a reset button, a temperature sensor, a miniature speaker with an amplifier, a microSD slot, a neopixel LED, an I2C port, and two ports for additional sensors. The last three ports also provide power and ground.

Mostly because of the extensive equipment, it's not surprising that the intelligent display has been very well received in the maker scene. Adafruit also designs the hardware, so you can adapt it to your own ideas. For example, if you need a better loudspeaker, just interrupt the defined conducting path and connect your own.

Besides the PyPortal, Adafruit also sells a matching minimalist display stand. However, both Adafruit and the community have already designed various cases that are available on popular 3D printing portals. You will find the stand from Figure 1, for example, on Thingiverse [2].

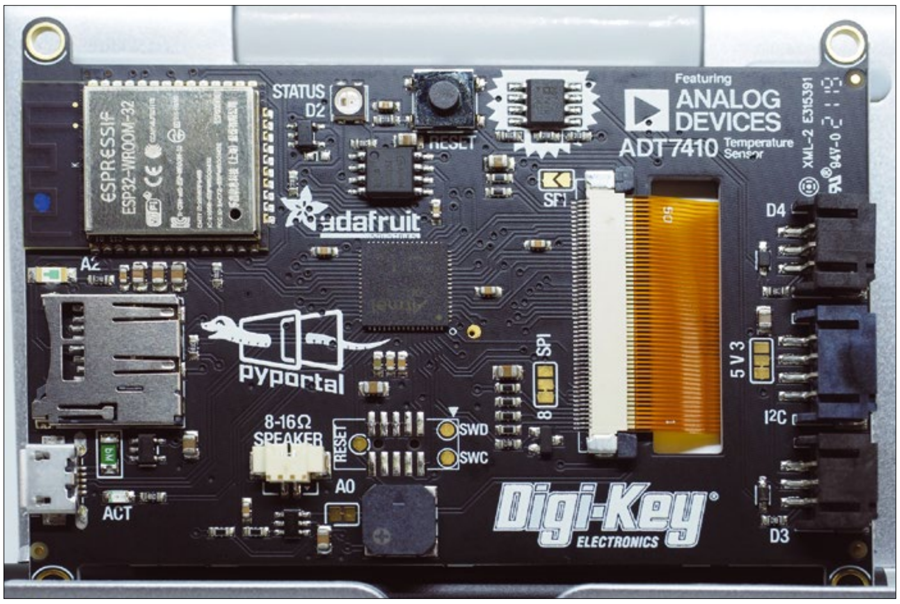


Figure 2: The back of the PyPortal features the Atmel CPU and the ESP32 chip, which handles WiFi and the computationally intensive TLS/SSL encryption protocol.

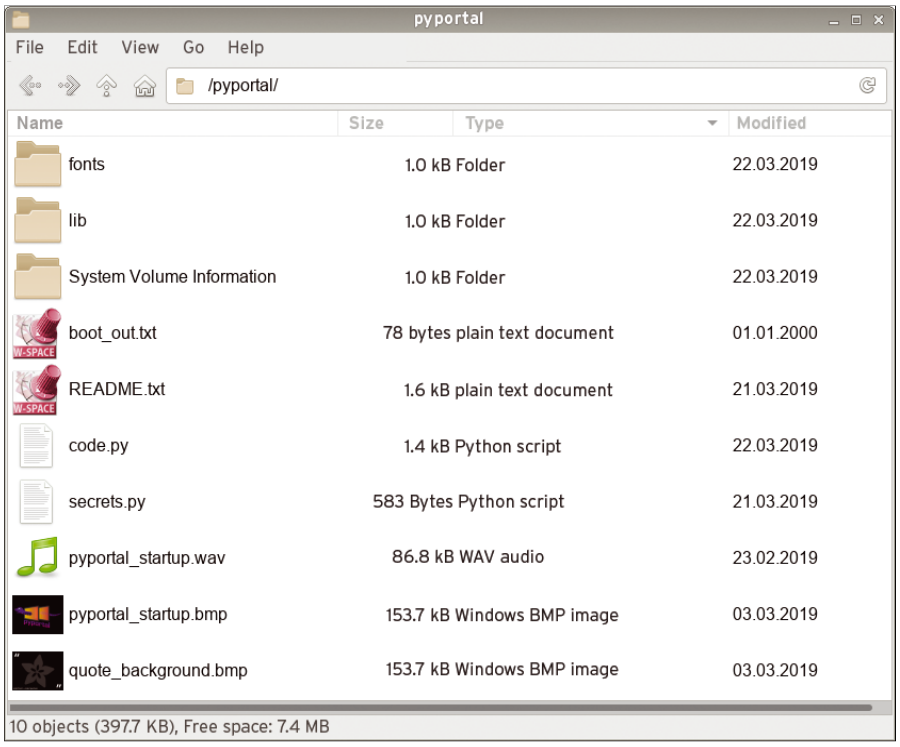
Getting Started

To put the display into operation, all you need to do is connect it to a 5V power supply with a micro-USB plug. CircuitPython runs on the Cortex-M4 processor; Adafruit has implemented a small example program, but it throws errors because the WiFi chip fails to connect to the home network without an SSID and password. Before you adapt the program, though, you will want to update the firmware.

Both firmware updates and program changes are handled over the USB cable, which you connect to a PC instead of a power supply for this purpose. Choose a cable that is not too long and of good quality; in particular, do not use a mere powercable. If you press the reset button once, the PyPortal reboots; if you press it twice, it switches to firmware update mode.

In both operating modes, the display provides its program memory as a drive. In normal operation, it goes by the name *CIRCUITPY*; in update mode the name is *PORTALBOOT*. On Linux, the device files known from USB storage media appear with a small difference: In operating mode, PyPortal announces itself as a partitioned mass storage device with the device name `/dev/sd<X>1`; in update mode it claims to be an unpartitioned stick named `/dev/sd<X>`.

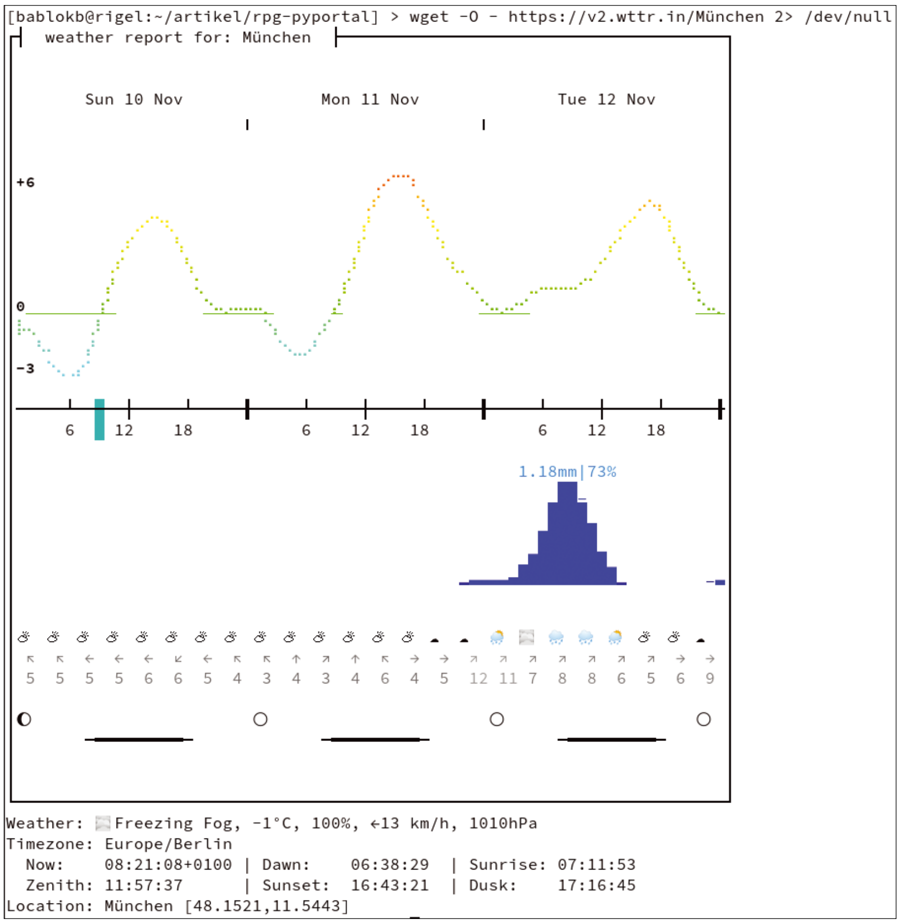
On Linux, first check to see whether the drive has been mounted automatically. If this is the case, eject it and remount it with the command:



Name	Size	Type	Modified
fonts	1.0 kB	Folder	22.03.2019
lib	1.0 kB	Folder	22.03.2019
System Volume Information	1.0 kB	Folder	22.03.2019
boot_out.txt	78 bytes	plain text document	01.01.2000
README.txt	1.6 kB	plain text document	21.03.2019
code.py	1.4 kB	Python script	22.03.2019
secrets.py	583 Bytes	Python script	21.03.2019
pyportal_startup.wav	86.8 kB	WAV audio	23.02.2019
pyportal_startup.bmp	153.7 kB	Windows BMP image	03.03.2019
quote_background.bmp	153.7 kB	Windows BMP image	03.03.2019

10 objects (397.7 KB), Free space: 7.4 MB

Figure 3: When connected to a Linux computer, PyPortal's memory appears to be a normal drive.



```
[bablokb@rigel:~/artikel/rpg-pyportal] > wget -O - https://v2.wttr.in/München 2> /dev/null
 weather report for: München

      Sun 10 Nov                 Mon 11 Nov                 Tue 12 Nov

+6

0

-3

     6    12    18          6    12    18          6    12    18

                                        1.18mm|73%

 5  5  5  5  6  6  5  4  3  4  3  4  6  4  5  12 11 7  8  8  6  5  6  9

Weather: Freezing Fog, -1°C, 100%, ←13 km/h, 1010hPa
Timezone: Europe/Berlin
  Now:    08:21:08+0100 | Dawn:    06:38:29  | Sunrise: 07:11:53
  Zenith: 11:57:37      | Sunset:  16:43:21  | Dusk:    17:16:45
Location: München [48.1521,11.5443]
```

Figure 4: The *wttr.in* project outputs the required weather data in ASCII format on the terminal.

Listing 1: Displaying an Image

```
import board
import displayio
from adafruit_bitmap_font import bitmap_font

# --------------------------------------------

def get_background(filename):
  bg_file = open(filename, "rb")
  background = displayio.OnDiskBitmap(bg_file)
  return displayio.TileGrid(background, pixel_shader=displayio.ColorConverter())

# --- main-loop   ---------------------------

display = board.DISPLAY
group   = displayio.Group()

# set background
group.append(get_background("clouds.bmp"))

display.show(group)

while True:
  time.sleep(60)
```

```
$ sudo mount -o sync /dev/sd<X> /mnt
```

The `sync` option ensures that all files are immediately written to the PyPortal. To install the firmware update, first download the current version, and then simply copy the file with the `.uf2` suffix to the drive. The board then reboots, the *PORTALBOOT* drive disappears, and the *CIRCUITPY* drive appears.

Details of the update and, more specifically, links to the current firmware and how to work with CircuitPython can be found in the Adafruit [3] documentation. At the end of the day, working with the PyPortal is no different from working with other CircuitPython boards.

Figure 3 shows the layout of the files in operation mode. The root directory contains the `code.py` file, which contains the Python code of the main program that runs in an infinite loop, and the `lib/` subdirectory contains libraries. Other files and directories depend on the application.

Weather Geek

Adafruit's sample program, as supplied with the product, retrieves a saying from a web server at regular intervals and shows it on the display. To log in to your WiFi network, the program needs the access credentials, which you enter in the `secrets.py` file. On closer inspection, the program turns out not to be very instructive, because it hides all the magic in the `PyPortal` class. This prompted me to develop a project of my own to illustrate how to display images and text.

The PyPortal is best suited for receiving and visualizing data. The program will query the weather from *wttr.in*. This slightly different kind of weather portal provides an ad-free weather report on the command line and is based on the Go program `wego`; fans of ASCII art will definitely get their money's worth. Figure 4 shows a deluxe version of its output. In a terminal window, you can call up this weather forecast for your location by typing `curl wttr.in`. Typing `curl wttr.in/:help` shows the syntax.

Wallpapering

The first thing to do is to set up some nice wallpaper on the screen. The PyPortal only supports bitmaps in 8-bit

Listing 2: Creating a Text Box

```
01 import board
02 import time
03 import displayio
04 from adafruit_bitmap_font import bitmap_font
05 from adafruit_display_text import label
06
07 # -------------------------------------------
08
09 def get_label(text,font,color):
10   text_area = label.Label(font, text=text, color=color)
11   (x,y,w,h) = text_area.bounding_box
12   text_area.x = 0
13   text_area.y = -y
14   return text_area
15
16 # --- main-loop   --------------------------
17
18 display = board.DISPLAY
19 group   = displayio.Group()
20 FONT    = bitmap_font.load_font("/DroidSansMono-16.bdf")
21 COLOR   = 0xFFFFFF
22
23 text="""Row1
24 Row2
25 Row3"""
26
27 group.append(get_label(text,FONT,COLOR))
28 display.show(group)
29
30 while True:
31   time.sleep(60)
```

format at 320x240 pixels. For the project, I trimmed a picture of clouds in Gimp accordingly and exported it in BMP format. The code from Listing 1 displays the image.

The `displayio` user interface library uses its own nomenclature. Other toolkits build and display trees of widgets, referred to as a `group` here. The program creates the top-level group in line 15. The `get_background()` subroutine (lines 7-10) creates a `TileGrid` (which itself is a group) from the BMP file.

The command in line 18 adds this group as the first subgroup and displays everything in line 20. The display determines the order of the subgroups: The earlier you add a subgroup, the further back it appears in the image, which is why the background needs to come first.

Text Output

The `Label` class is used for text output. Again, the terminology is confusing – a label is usually a short caption. In the `displayio` nomenclature, however, `Label` denotes a multiline text box. The library does all the work transparently in the constructor (Listing 2, line 10). The constructor creates a label from the text that is passed in, evaluates embedded line breaks, and takes care of suitable line spacing.

However, the absolute alignment of the entire text box is not very intuitive. If you simply output the label, you will only see half of the content: The `x` and `y` coordinates of the text box refer to the center of its left edge, which would then end up in the upper left corner of the display. Lines 11-13 correct the offset of the label so that the text output ends up where you expect it to.

If you want to output several pieces of text in different colors or sizes, you cannot do this with just one label. The dimensions of the bounding box in line 11 help you achieve the right layout.

Fonts

The text output always uses bitmap fonts, with all their inherent advantages and disadvantages. Bitmap fonts contain mini-images of all characters, which avoids the time-consuming rendering of letters. Because the character size is fixed, you need a separate font for each size. More powerful systems, where the rendering overhead is not important, therefore use more space-saving character set formats. As a result, hardly any bitmap fonts are available for download on the Internet.

Fortunately, you can get the font conversion program FontForge through your system's package manager, or you can download it from the FontForge homepage. By following the instructions found online [4], in just a few mouse clicks you have created a bitmap font of your choice. To display the ASCII graphics of the weather output correctly, I chose a monospace font.

Data from the Web

The only thing missing after the wallpaper text is the Internet connection, so you acquire the weather data. The decisive commands are shown in line 11 of Listing 3, which imports the SSID and the WiFi password from the `secrets.py` file. The code in lines 14-21 then establishes the connection with the ESP32 chip.

Once the connection is up, you can send standard HTTP requests like `GET` and `POST` and evaluate the server response – typically by HTML, but with JSON to query data servers. In contrast, *wttr.in* returns plain text (Listing 3, lines 28 and 30), which the program outputs directly after converting it to a label (Figure 5). I adapted the URL in line 27 to suit the small PyPortal, so the output only shows the current weather data instead of a detailed multiple-day forecast. Change the city name before the question mark in the URL for your location, and use `lang=en` for English output.

With fewer than 50 lines of code, the display shows the current weather data, but the program still lacks the ability to query the weather report regularly. To do this, you need to add a `get()` command to the infinite loop at the end of the program. Also, you might want to display the current time and the indoor temperature from the built-in sensor in the PyPortal.

Unfortunately, the temperature sensor in the PyPortal turns out to be a flop. The display backlighting heats it up, so it delivers useless values. As a consequence, Adafruit got rid of this sensor in the PyPortal Pynt and Titano variants. A better alternative would be to

Listing 3: Getting the Data

```
import board
import busio
from digitalio import DigitalInOut
import time
import neopixel
import displayio
from adafruit_esp32spi import adafruit_esp32spi, adafruit_esp32spi_wifimanager

# ---------------------------------------------

from secrets import secrets  # file secrets.py

def get_wifi(secrets):
  esp32_ready = DigitalInOut(board.ESP_BUSY)
  esp32_gpio0 = DigitalInOut(board.ESP_GPIO0)
  esp32_reset = DigitalInOut(board.ESP_RESET)
  esp32_cs    = DigitalInOut(board.ESP_CS)
  spi         = busio.SPI(board.SCK, board.MOSI, board.MISO)
  esp         = adafruit_esp32spi.ESP_SPIcontrol(spi, esp32_cs, esp32_ready, esp32_reset, esp32_gpio0)
  status_rgb  = neopixel.NeoPixel(board.NEOPIXEL, 1, brightness=0.2)
  return adafruit_esp32spi_wifimanager.ESPSPI_WiFiManager(esp, secrets, status_rgb)

# --- main-loop   -----------------------------

connection = get_wifi(secrets)
try:
  response = connection.get("https://wttr.in/München?AT0&lang=en")
  text = response.text
except:
  text = "No data received"
group.append(get_label(text, FONT, COLOR))
display.show(group)

while True:
  time.sleep(60)
```

attach an alternative device (e.g., a BME280 or LM75) with a short cable to the I2C interface.

An extended version of this sample program is on GitHub [5].

Better than the Raspberry Pi?

The sample project has shown at least some of the PyPortal's capabilities. The question arises as to how the intelligent Adafruit display compares with a Pi Zero plus a miniature display.

The biggest benefit the PyPortal offers is its compact design. The processors, components, and connections on the rear are unlikely to wear out. Conversely, the Raspberry Pi is attached in a fairly precarious way to the socket array or HDMI port on a typical miniature screen. In the former case, the screen usually blocks pins that it doesn't need, so connecting additional sensors then means purchasing a multiplexer board.

Unlike the Raspberry Pi, the PyPortal comes with a tiny speaker for sound output. It is more suitable for warnings and signal tones than for listening to music, but the Pi Zero lacks this feature completely.

Regardless of the hardware, the intelligent display is far easier to set up than the Pi Zero and a small screen. Tinkering with the `/boot/config.txt` file with special overlays or HDMI parameters is not necessary. To switch it off, all you have to do is pull the plug, because there is no operating system that needs to be shut down cleanly first.

The PyPortal's current draw is about 200mA when the display is on and about 70mA otherwise. Therefore, the display is not suitable for battery operation. Even in continuous operation, though, it is unlikely to increase your electricity bill that much. A display of exactly the same size for the Raspberry Pi was not available for comparison measurements, but with a 4-inch screen from Waveshare, the combination clocked up to 260mA. The difference of 60mA between the PyPortal and Pi Zero including display can be measured, but in practice it hardly plays a role.

Of course, PyPortal does not always give you the better solution because many of its advantages turn into disadvantages

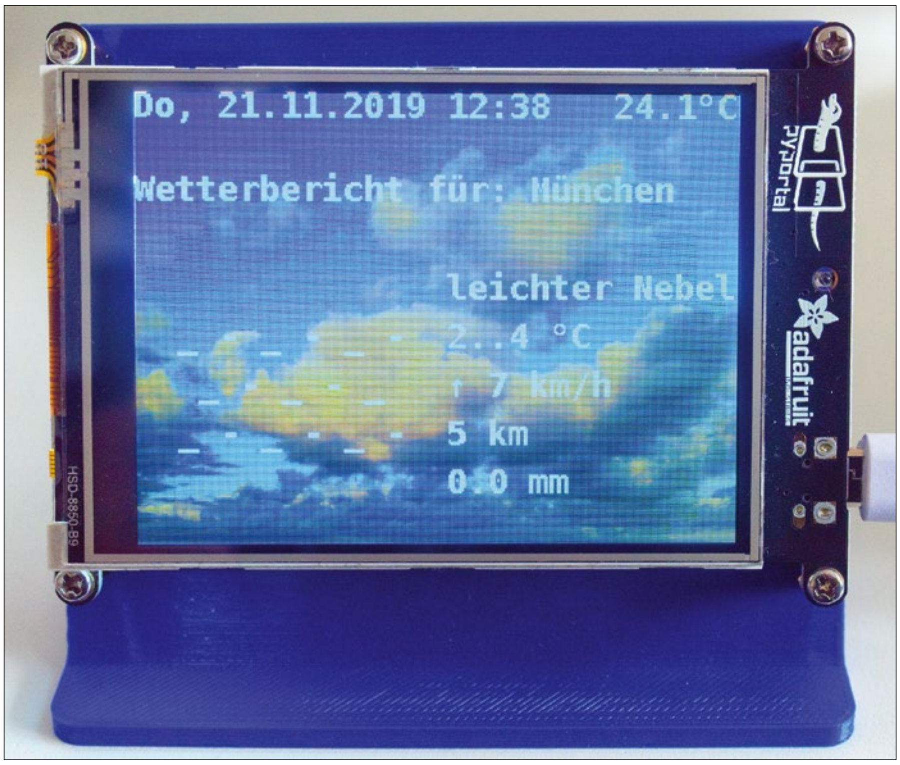

Do, 21.11.2019 12:38 24.1°C
Wetterbericht für: München
leichter Nebel
2..4 °C
↑ 7 km/h
5 km
0.0 mm

Figure 5: The weather forecast for Munich, as requested from *wttr.in*, shows light fog. A picture of clouds was adapted for the background.

in certain applications. If you need considerably more screen surface or better resolution, there is no escaping the Raspberry Pi, which also offers a faster CPU, more RAM, and multitasking. These more extensive resources also give you scope for more demanding and complex programs. When designing graphical user interfaces, especially, toolkits with layout management offer more possibilities for the developer than the PyPortal's `displayio` library.

Conclusions

As the lively community that has grown up around the PyPortal shows, Adafruit has its finger on the users' pulse with this display. Thanks to the fixed form factor, everyone programs the same hardware, and open source promotes the exchange of ideas and solutions. If you want to discover more details about the PyPortal or are looking for suggestions, you will find the guides [6] to be a comprehensive resource. In many projects, teaming the Raspberry Pi and the PyPortal also makes sense. The Pi handles the computationally intensive and memory-intensive work, and the PyPortal presents the results. ■■■

Info

[1] Buying the PyPortal display: *https://www.adafruit.com/product/4116*

[2] Housings and brackets: *https://www.thingiverse.com/search?sort=relevant&q=PyPortal&type=collections*

[3] Main documentation: *https://learn.adafruit.com/adafruit-pyportal*

[4] Generating bitmap fonts: *https://learn.adafruit.com/custom-fonts-for-pyportal-circuitpython-display*

[5] Sample project for this article: *https://github.com/bablokb/pyportal-wttrin*

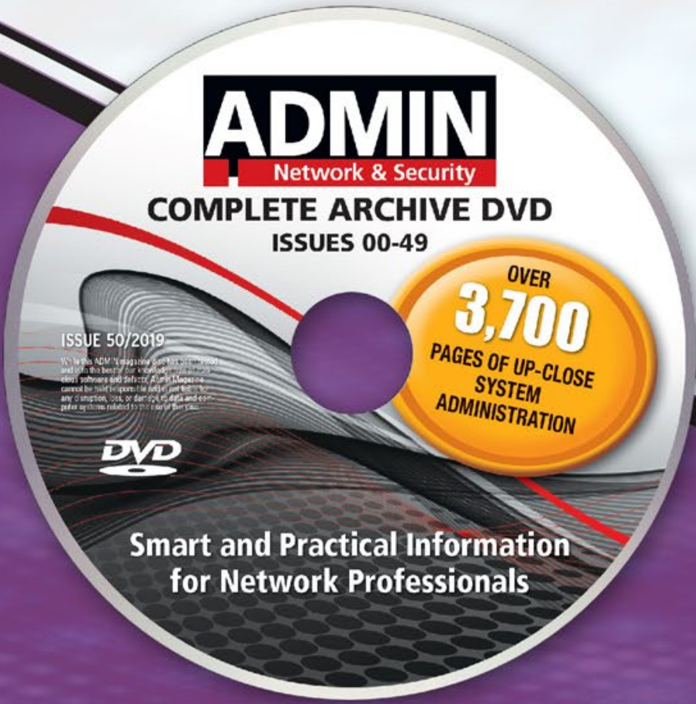
[6] All PyPortal guides: *https://learn.adafruit.com/products/4116/guides*

Author

Bernhard Bablok works at Allianz Technology SE as an SAP HR developer. When he's not listening to music, cycling, or walking, he deals with topics related to Linux, programming, and small computers. He can be reached at *mail@bablokb.de*.

■■■

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


ADMIN Network & Security
COMPLETE ARCHIVE DVD
ISSUES 00-49
OVER 3,700 PAGES OF UP-CLOSE SYSTEM ADMINISTRATION
ISSUE 50/2019
DVD
Smart and Practical Information for Network Professionals

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

THE COMPLETE
UBUNTU
user
ARCHIVE
DVD ROM
Searchable DVD
Content Available in Both HTML and PDF Formats
Over 3,000 PAGES! 7 GREAT YEARS OF UBUNTU USER

MakerSpace

Build your own kitchen timer with a dual alarm

Kitchen Helper

A simple kitchen helper with two timers assists budding chefs in coping with dishes that are unlikely to be ready at the same time. *By Bernhard Bablok*

Not all the recipes you cook likely need to be ready at the same time, especially if everyone in the household has their own idea of how long you need to boil eggs for breakfast. A simple dual timer solves the problem.

This project uses a Pi Zero and some cheap components sold by the usual mail order companies for a few euros, dollars, or pounds. In principle, it is possible to build the project from scratch on a breadboard without any soldering, which makes it perfect as a starter project for your own experiments.

On the Pi Zero you need to install Raspbian "Buster" up front; the Lite version will do the trick. The only additional package you need to install for the time being is wiringPi [1], which uses a numbering system different from the header pin and GPIO systems. If you want to work with a graphical user interface, you could switch to a Raspberry Pi 3 or 4, but these platforms are oversized for this use case.

Strictly speaking, even the Pi Zero is overkill because, in principle, any microcontroller could control an alarm clock. The advantage of the Raspberry Pi is that software implementation is faster and easier: I use Python in this project. Before you move on to the software for the project, you first need to wire the hardware.

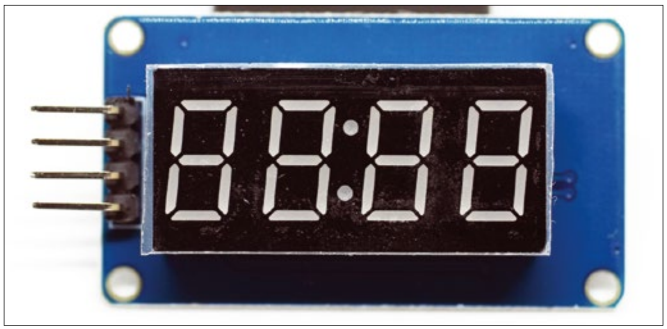


Figure 1: A typical seven-segment display.

Mini Displays

Segment displays based on the TM1637 driver are ideal for displaying times (Figure 1). Each digit can be displayed with seven segments, and suitable displays are easily found by searching for "7-segment display." The monochrome displays are inexpensively available in red, white, blue, yellow, and green. They are available with a colon in the middle (clock type) or with one decimal point per digit.

Many sellers (especially on eBay) do not always provide the correct data. Often you will find a description of the clock type combined with an image of the decimal point display. Other sellers show both types to protect themselves against complaints relating to wrong delivery. Make sure you take a close look.

The connector pins are another potential source of concern. Pre-soldered specimens, as shown in Figure 1, are useful for experiments on the breadboard, but for installation in a housing, pins soldered from the back would be more useful. Finding the perfect device is not that easy.

TM1637 displays require two lines (*CLK* and *DIO*, or clock and data) in ad-

Lead Image © besjunior, 123RF.com

dition to power and ground. It is a lean I2C protocol that works on any pin. The device can handle 5V of input voltage, but the 3.3V on the Raspberry Pi are perfectly OK.

In this project, the displays are deployed on GPIO6 and GPIO12 (*CLK*) and GPIO13 and GPIO16 (*DIO*) on the left side of the display. Figure 2 shows the complete assembly with wiring for orientation.

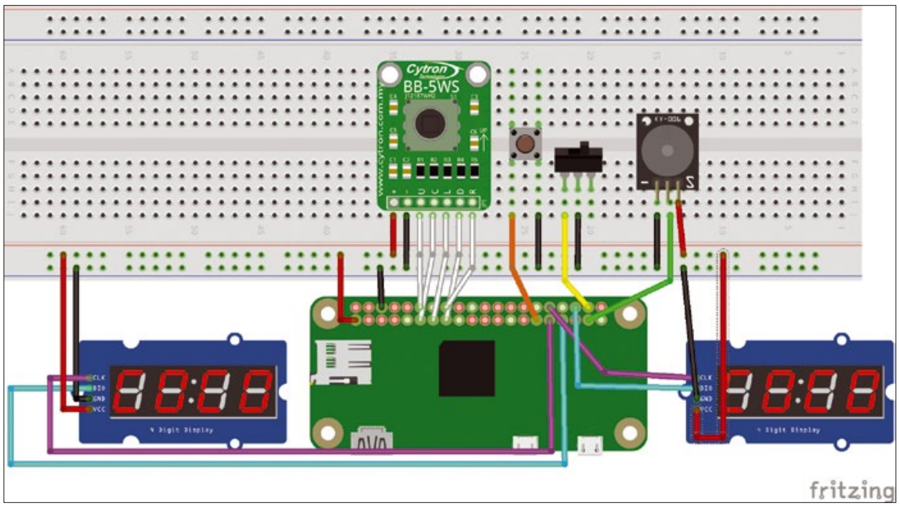


Figure 2: Fritzing view of the wiring.

Sound the Alarm

A buzzer serves as an alarm signal. You want to use an active buzzer; for a passive version, you would have to generate the sound yourself as PWM signals. Buzzers are typically available with two or three pins. In the first case you just need to apply a voltage (set the GPIO to High); in the alternative case, you have a control pin in addition to voltage and ground. For the versions I used for this example, I had to set the control pin to Low to make the buzzer go off.

The alarm clock program expects the control pin on GPIO26. To check this functionality, enter:

```
$ gpio -g mode 26 out
$ gpio -g write 26 0
$ gpio -g write 26 1
```

The first line switches GPIO26 as the output pin, and the second line activates the buzzer. The third line switches it off again. At startup time, GPIO26 is an input pin and you might hear a short sound. If this annoys you, a low pullup resistor will prevent this happening.

Toggle

To edit the alarm times, you need to be able to switch between the two displays. To do so, you can use a slider, which is a button with two states. Of its three connectors, the central is grounded, and depending on the position of the switch, either the left or the right connector is grounded.

In this project, only the left connector GPIO20 and ground are wired. Also, the GPIO20 internal pullup is switched on. While the switch is on the right, GPIO20 is High, but otherwise Low. To test the wiring, enter:

```
$ gpio -g mode 20 in
$ gpio -g mode 20 up
$ gpio -g read 20
```

Repeat the command in the third line for different switch positions; depending on the position, the `read` command outputs a *0* or *1*.

Buttons

The project uses a five-way switch to set the alarm times (Figure 3). One press toggles between setup and standby, left/right changes the position in the row of digits, and up/down changes the value. Although the vendor recently discontinued the switch I use in this example, you can find similar models – even with two additional buttons – on eBay at a low cost (Figure 4). Search for *navigation button*. The Cytron switch has a pull-up resistor and a debouncing capacitor for each connector that are often missing from eBay products.

Because the Raspberry Pi has configurable pullups, the lack of resistors is easy to remedy. You can either upgrade the capacitors (100pF) yourself or simply debounce them in the software (see below). My eBay button was even presoldered, and I was able to build the project on a breadboard without any soldering.

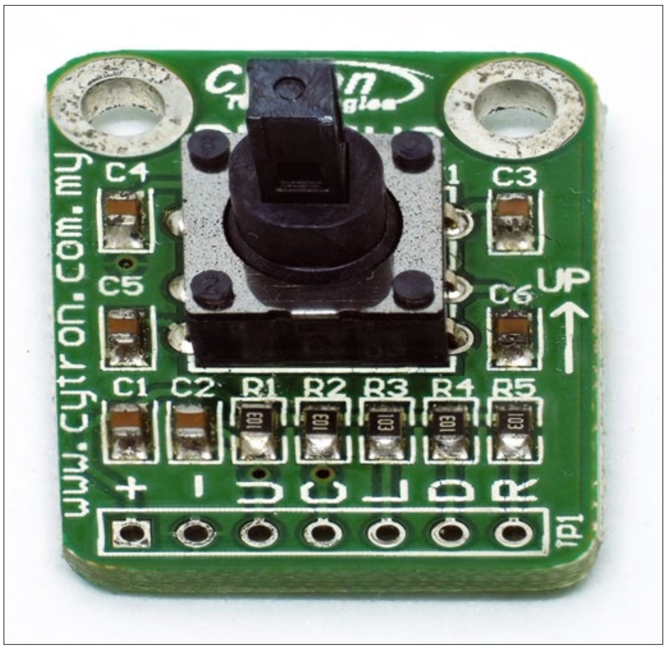


Figure 3: A five-way switch by Cytron.

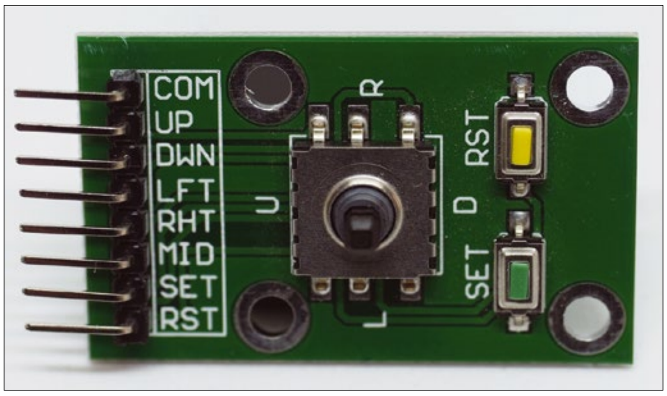


Figure 4: An alternative no-name button on eBay.

Table 1: Pin Assignments

GPIO	Wiring
17	C (command/press)
18	U (up)
22	D (down)
27	L (left)
23	R (right)

The Raspberry Pi needs a separate pin for each direction of the navigation switch. The pin assignments can be found in Table 1. Check to see that everything is wired correctly (Listing 1). The test waits for the corresponding direction to be pressed in turn and then outputs a message.

To do this, line 6 uses the `wfi` function, which stands for `wait for interrupt` and means that the program waits until the pin triggers an interrupt. In this case, only the transition from High to Low is of interest (i.e., `falling`; additionally, you can have `rising` or `both` signals).

Besides the five-way button, a large, easy-to-use button starts and stops the alarm timer. The button is connected to GPIO5, and you can test it as described before. The Raspberry Pi provides the pullup for this button, and debouncing is done by software.

Alarm Timer Software

If you are looking for a challenge, you can write the software yourself. To make things easier, download the program from my GitHub repository [2]. If you have not already done so, install the *git* package and then set up the software with:

```
$ git clone https://github.com/↩
  bablokb/doubleclock.git
$ cd doubleclock
$ sudo tools/install
```

The program code is in the `/usr/local/sbin/doubleclock.py` file. The installation program sets up a systemd service that starts the alarm timer at boot time.

For some initial tests, however, it is advisable to disable the service right away and start the program manually with:

```
$ sudo systemctl disable ↩
  doubleclock.service
$ doubleclock.py
```

The advantage of the manual call is that you can see potential program errors on the screen. Ideally, the two display segments now fill up with zeros. You can terminate the program at any time by pressing Ctrl+C.

The structure of the program is simple: The main thread updates the displays twice per second. Asynchronously, methods react to control commands. For example, the `do_left()` method processes five-way switch presses to the left.

Once the alarm timer is running, another thread counts down the remaining time. If the alarm timer rings, a third thread takes care of the buzzer and elicits its rhythmic sounds. All of this is achieved with a little Python wizardry contained in about 200 lines of code plus comments.

Adaptations

Depending on the available hardware, the next step is to customize the program. Listing 2 shows a section of the switch configuration. The variable names that start with `PIN_` are constants; the corresponding GPIO numbers are assigned early in the program.

Except for the buzzer, the program defines all the pins as input (lines 64-71). If the switch module you use does not have pullups, you will need to add lines 64-68, as per line 69. Lines 73-79 each assign a method to the pins that is executed by the program when the user presses a button.

The last parameter in line 79 is the debounce time – 200ms here. The other buttons have capacitors for debouncing and do not need this delay. If the device you use does not have capacitors, then add the third parameter to lines 73-77, as well.

No further adjustments should be needed. If you want to rebuild the project with just one display, connect the left display (GPIO6/GPIO13), do without the slider, and connect GPIO20 to ground. No changes to the software are required.

Listing 1: Testing the Button

```
01 #!/bin/bash
02 echo -e "\nOrder: set, up, down, left, right\n\n"
03 for pin in 17 18 22 27 23; do
04   gpio -g mode "$pin" in
05   gpio -g mode "$pin" up
06   gpio -g wfi "$pin" falling
07   echo "GPIO$pin is now LOW"
08 done
```

Listing 2: Switch Config Snippet

```
[...]
64 GPIO.setup(PIN_PUSH,     GPIO.IN)
65 GPIO.setup(PIN_UP,       GPIO.IN)
66 GPIO.setup(PIN_DOWN,     GPIO.IN)
67 GPIO.setup(PIN_LEFT,     GPIO.IN)
68 GPIO.setup(PIN_RIGHT,    GPIO.IN)
69 GPIO.setup(PIN_SLIDER_L, GPIO.IN,  pull_up_down=GPIO.PUD_UP)
70 GPIO.setup(PIN_START,    GPIO.IN,  pull_up_down=GPIO.PUD_UP)
71 GPIO.setup(PIN_BUZZER,   GPIO.OUT, initial=1-PIN_BUZZER_ON)
72
73 GPIO.add_event_detect(PIN_PUSH,     GPIO.FALLING, self.on_push)
74 GPIO.add_event_detect(PIN_UP,       GPIO.FALLING, self.on_up)
75 GPIO.add_event_detect(PIN_DOWN,     GPIO.FALLING, self.on_down)
76 GPIO.add_event_detect(PIN_LEFT,     GPIO.FALLING, self.on_left)
77 GPIO.add_event_detect(PIN_RIGHT,    GPIO.FALLING, self.on_right)
78 GPIO.add_event_detect(PIN_SLIDER_L, GPIO.BOTH,    self.on_slider)
79 GPIO.add_event_detect(PIN_START,    GPIO.FALLING, self.on_start, 200)
[...]
```

Regardless of the necessary adjustments, the code allows you to configure various parts to suit your preferences. For example, the audible alarm signal from the left alarm timer stops automatically after 10 seconds, whereas the right alarm continues to run indefinitely. You can even control the maximum brightness of the display (values between `0` and 7). Last but not least, you are free to choose the pin assignments.

If you wired and set up everything correctly, the commands

```
$ sudo systemctl enable ⮐
  doubleclock.service
$ sudo systemctl start ⮐
  doubleclock.service
```

enable and start the service.

System Optimization

The project is complete in principle, but you can take advantage of a few optimizations to round everything off. If you have a spare power outlet and want to keep the alarm timer running all the time, you don't need to worry about booting the system, but if you only turn on the alarm timer when needed, you will soon notice that booting takes a long time and the wait can be annoying. Fortunately, a couple of tricks will reduce the wait significantly [3].

Another optimization is related to the shutdown behavior. The alarm timer does not have a control for this, but you could retrofit one in the form of an additional button. Of course, in the heat of the moment, the chef might forget more than just the salt. It makes more sense to set up the system to be read-only, which means Linux then no longer writes to the SD card. Instead of shutting down, you can simply pull the plug [4].

In the special case of this example, a much simpler method also works, but only because the program does not even write temporary files and all the other programs on the Raspberry Pi are irrelevant: Simply comment out all lines in the `/etc/fstab` file. The kernel will still mount the root filesystem at startup, but in read-only mode, and the boot filesystem is unnecessary for normal operation.

Besides working under the hood, you also want the appearance to be pleasing. For the alarm timer, an attractive 3D-printed case is a good choice, because a breadboard with jumper cables is not something you would want in your kitchen. Because the atmosphere in the kitchen tends to be quite humid, you will want to use PETG instead of the more common PLA [5]. I already have a case in the making, so it's worth taking a look at the project [6]. By the time this article is published, it might even be finished.

Don't underestimate the time needed to get from the breadboard to the finished alarm clock that you would want to see in the kitchen every day. Four modules are connected to 3.3V, but the Raspberry Pi offers only two 3.3V pins. Without an additional board, nothing will work. You also need room for the cables without having an oversized housing.

Another question is how to fasten the case while leaving at least the power connection accessible from outside. Mounting the smaller components, such as the single switch or the slider, is by no means trivial. They need a firm hold, because in everyday life, they will have to withstand forces perpendicular to the housing.

Conclusions

Your own projects with cheap hardware can be prototyped with just a breadboard and a few jumper cables. The software usually takes a little more work, but thanks to the many examples online, the programming effort can be minimal.

Starting with this example as a jumping off point, you can realize your own ideas. If the alarm timer is not running, it would be a good idea to show the time and date on the two displays. I would be happy to field any corresponding pull requests. Another idea would be to convert the alarm clock into an electronic chess clock, where two buttons alternately start the timers. The possibilities offered by the Raspberry Pi are (almost) unlimited. ■■■

Info

[1] wiringPi: *http://wiringpi.com/download-and-install/*

[2] Dual timer project: *https://github.com/bablokb/doubleclock*

[3] "Fast boot with Raspberry Pi" by Himesh Prasad: *https://himeshp.blogspot.com/2018/08/fast-boot-with-raspberry-pi.html*

[4] "Make your Raspberry Pi file system read-only (Raspbian Buster)" by Andreas Schallwig, *Medium*, 23 September 2019, *https://medium.com/swlh/make-your-raspberry-pi-file-system-read-only-raspbian-buster-c558694de79*

[5] "PETG vs PLA: The Differences – Simply Explained" by Lamin Kivelä, *All3DP*, 26 January 2020, *https://all3dp.com/2/petg-vs-pla-3d-printing-filaments-compared/*

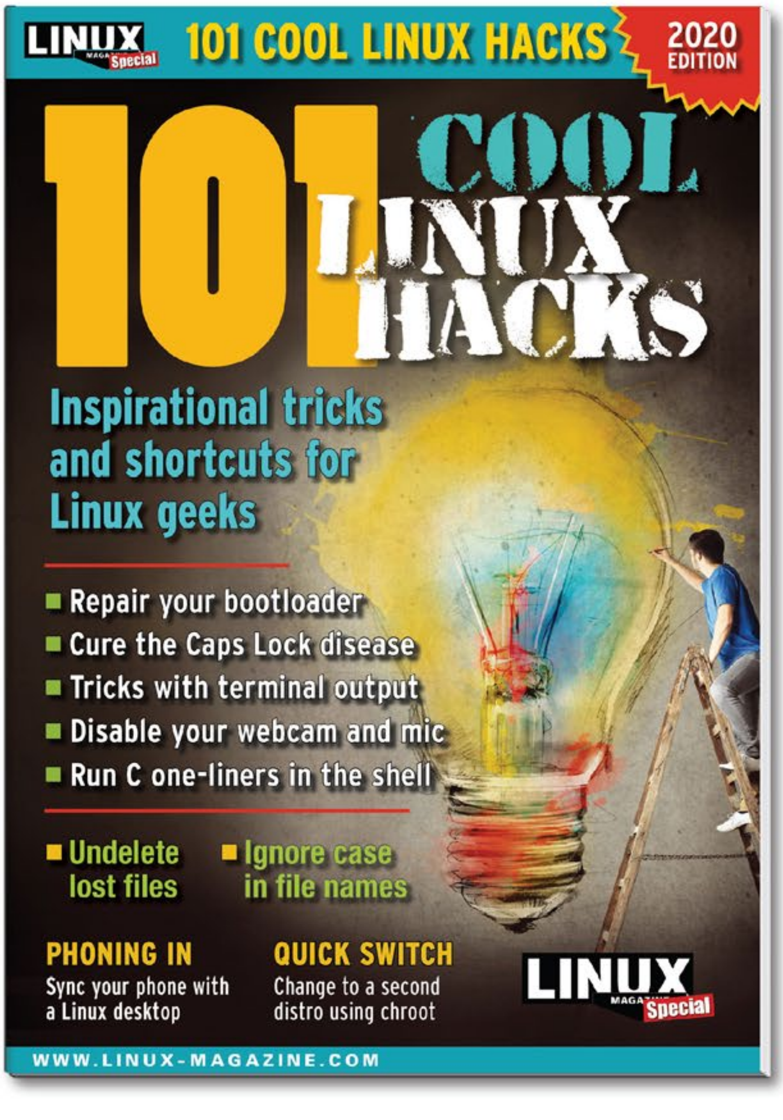
[6] Tinkercad files: *https://www.tinkercad.com/things/8dUa6ugfUpZ*

Author

Bernhard Bablok works at Allianz Technology SE as an SAP HR developer. When he's not listening to music, cycling or walking, he deals with topics related to Linux, programming and small computers. He can be reached at *mail@bablokb.de*.

SHOP THE SHOP
shop.linuxnewmedia.com

GET PRODUCTIVE WITH
101 LINUX HACKS


LINUX MAGAZINE Special
101 COOL LINUX HACKS
2020 EDITION
101 COOL LINUX HACKS
Inspirational tricks and shortcuts for Linux geeks
Repair your bootloader
Cure the Caps Lock disease
Tricks with terminal output
Disable your webcam and mic
Run C one-liners in the shell
Undelete lost files
Ignore case in file names
PHONING IN
Sync your phone with a Linux desktop
QUICK SWITCH
Change to a second distro using chroot
LINUX MAGAZINE Special
WWW.LINUX-MAGAZINE.COM

Improve your Linux skills with this cool collection of inspirational tricks and shortcuts for Linux geeks.

- **Undelete lost files**
- **Cure the caps lock disease**
- **Run C one-liners in the shell**
- **Disable your webcam and mic**
- **And more!**

ORDER ONLINE:
shop.linuxnewmedia.com/specials

Is the desktop really just a metaphor? You bet it is. The software running on your computer *doesn't really look like anything*, and that icon you click on inhabits the familiar form of a printer or a camera just to keep you from freaking out. All these metaphorical graphic aids have to come from somewhere, and if you work with Linux, they often come from Gimp. This month we present a Gimp workshop that takes you through the steps of creating a graphical skin for a media player application.

Also in this month's LinuxVoice, we introduce you to Waterfox, a clone of the Mozilla Firefox browser designed for privacy and support for legacy add-ons. We also show MystiQ, a handy tool for converting multimedia file formats.

Image © Olexandr Moroz, 123RF.com

LINUXVOICE

Doghouse – Old Systems 72
Jon "maddog" Hall
As unemployment claims surge, US computer systems are straining under the increased load. maddog weighs in on COVID-19 and COBOL.

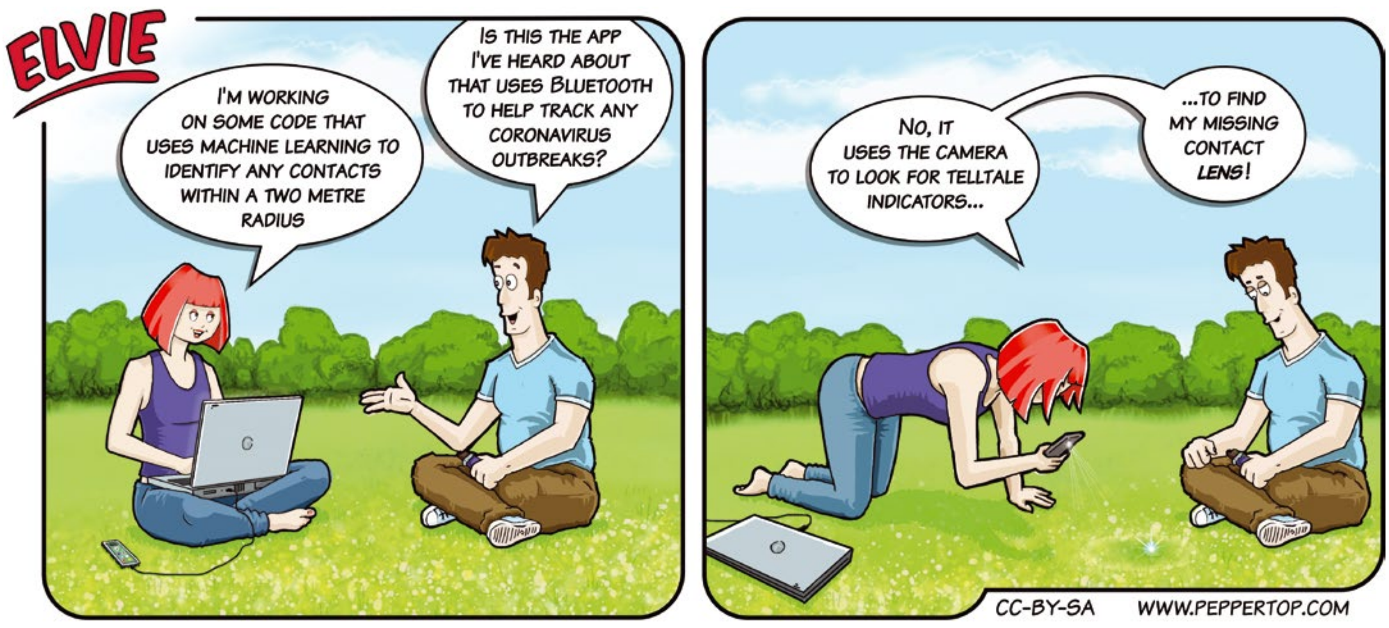
Waterfox 74
Erik Bärwaldt
This fork of the Mozilla Firefox browser is designed for greater speed and privacy, as well as compatibility with older add-ons.

Gimp Workshop: Media Player Skin 76
Gaurav Nawani
Forget the photos – Gimp is also a great tool for graphic design. This workshop shows how to use Gimp to create a simple visual image for a multimedia player application.

MystiQ 80
Erik Bärwaldt
Working with multimedia files usually involves converting from one format to another. MystiQ is a fast and easy tool for file conversion.

FOSSPicks 84
Graham Morrison
This month Graham looks at Horizon EDA, ksnip, Pi-hole 5, and more!

Tutorials – LÖVE 90
Paul Brown
To create an action-packed game with LÖVE, you'll need to overlay images to "physical" objects, detect collisions, and get input from the keyboard or mouse.


ELVIE
I'M WORKING ON SOME CODE THAT USES MACHINE LEARNING TO IDENTIFY ANY CONTACTS WITHIN A TWO METRE RADIUS
IS THIS THE APP I'VE HEARD ABOUT THAT USES BLUETOOTH TO HELP TRACK ANY CORONAVIRUS OUTBREAKS?
NO, IT USES THE CAMERA TO LOOK FOR TELLTALE INDICATORS...
...TO FIND MY MISSING CONTACT LENS!
CC-BY-SA
WWW.PEPPERTOP.COM

MADDOG'S DOGHOUSE

As unemployment claims surge, US computer systems are straining under the increased load. In this column, maddog weighs in on COVID-19 and COBOL. BY JON "MADDOG" HALL

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

Aging Computer Infrastructure

The first language I ever programmed was Fortran, and I learned to program it in 1969 by reading a book and practicing on an IBM 1130 computer that ran one job at a time and had basically no operating system. You linked the device drivers for the hardware into your program and effectively booted your program, not the operating system. Later in life, I would tell my students that we were not doing "Computer Science," but "Computer Black Magic."

Few systems were "networked," or if they were, they were dial-up networks to computers that held "bulletin boards" for exchanging programs and data. Music was on vinyl, and graphics were ASCII art printed on line printers (for the most part).

I am writing this today because of a YouTube video by Russell Brandom(*https://www.youtube.com/watch?reload=9&v=Ox_Wm6X-Qnxl*), who criticized the US government for still using COBOL on certain systems. He said that these systems were not up to the huge number of unemployment claims currently being filed, whereas companies like Netflix can scale to meet the large number of demands on them "even during the coronavirus outbreak." He inferred that it was the "fault" of COBOL.

Then one of my Facebook fans asked for my comment on a comparison between the "government's system" and Netflix.

Here are a few caveats before I even start.

First, unemployment claims are handled on a state-by-state basis, so their systems, software, and rules are different between states.

Second, I have not studied any of these systems, at all, nor have I spent a lot of time closely studying Netflix.

Third, I know that in many government departments it is hard to get money to update systems that are "working fine," particularly when you cannot prove a cost savings in doing the job you are already doing.

So with those three caveats, I started drawing comparisons.

Probably the systems to enroll unemployment requests were written some time ago, and COBOL was a fairly good language in which to write rules-based transactions.

The systems might typically have input during business hours, between 9am and 5pm (notice my use of this quaint designation for time) five days a week and not on federal or state holidays.

When unemployment is low, meeting the demands for new enrollment would be "easy," and when unemployment started to go up in peak times (mild recession, seasonal changes), people could work after hours and on weekends to meet the demands.

Typically this work might be done on highly dependable mainframe computers, designed with the hot-swap and redundant capabilities in mind. Even if the mainframe were to fail dramatically, the machine could be fixed relatively quickly and the processing of claims would continue.

In peak times, enrollment might be delayed a day or two, and people might not even notice it.

To plan for the highest peak loads over a large period of time would not be cost effective, since that computer would probably not be shared with any other non-departmental load (such as the agriculture department).

Over the past couple of years the unemployment rate has been very low. In "normal" years, it might be 3.5 to 4.5 percent; if unemployment goes to zero percent (everyone has a job), this makes is hard for employers to find new employees.

In the recession of 2008, the unemployment rate went up to 10 percent of the total workforce, but that happened over several months. As people lost their jobs, they would apply for unemployment funds. The systems could handle it, because, at the *peak* of that recession, nationwide claims were less than one million claims per week.

In this pandemic, the unemployment rate went from 3.5 percent to by some estimates over 25 percent of the population, with over 30 million claims in a six week period or between 5-6 times what the peak was in the recession of 2008, and 10-12 times what the rate was "normally."

Netflix (and many other online services) have different demands. They are meant to be available "24 by 7." Typically as demands for service come in, they are distributed to a large group of servers, so servers may be added or subtracted (for maintenance) as needed.

Services do see "peak loads," but they tend to be focused more around holidays and events. In any case, while Netflix is probably experiencing higher than average loads, I doubt very sincerely that they are 10-12 times their "normal" peak load, or even 5-6 times.

Netflix also has other techniques for handling loads. They can offer only SD movies instead of HD movies. They can stream at lower speeds.

COVID-19 is bringing out many lessons. When we talk about repairing our "aging infrastructure" in the US, we normally talk about highways and bridges. Perhaps we should also talk about our aging computer systems. ■■■

Shop the Shop shop.linuxnewmedia.com

Want to subscribe?

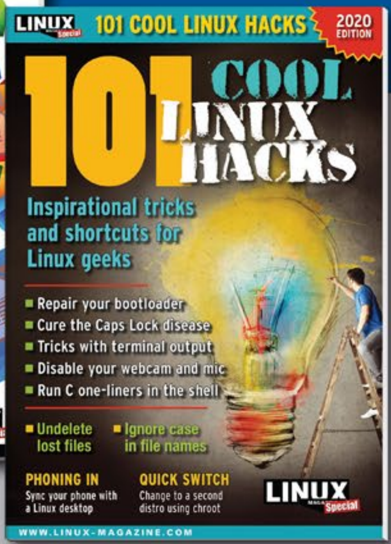
Searching for that back issue you really wish you'd picked up at the newsstand?

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

▶▶ shop.linuxnewmedia.com ◀◀

DIGITAL & PRINT SUBSCRIPTIONS

SPECIAL EDITIONS



Testing Waterfox, a Firefox alternative

Browsing Options

Waterfox, a fork of the Mozilla Firefox browser, is designed for greater speed and privacy, as well as compatibility with older add-ons. We consider two versions of the Waterfox browser and what they have to offer. BY ERIK BÄRWALDT

Mozilla's Firefox web browser has been massively popular in the past. Although for a long time the browser was one of the most frequently used web navigators, Google's Chrome browser and its open source counterpart Chromium have now outpaced it – not least because of faster response times.

Firefox also lost some of its popularity because of changes to the browser. The integration of services such as Pocket, interface changes, and the fact that telemetry data are collected have left many users concerned about their privacy and feeling restricted in terms of usage options. Also consider that for many privacy-conscious users, Chrome or Chromium are not an alternative because of the integration of many Google services with these browsers.

All of this makes Waterfox an interesting alternative. Waterfox is a fork of the Mozilla browser that claims to have removed all unnecessary elements from the software, especially unnecessary data collection [1]. Waterfox also lets you continue using NPAPI plugins that Firefox has not supported for a long time.

Getting Started

Two versions of Waterfox are available for download from the project website, Waterfox Current and Waterfox Classic. While the Current version implements new technologies and is therefore more suitable for fans of bleeding edge software, the Classic version uses older browser technologies, including support for NPAPI plugins.

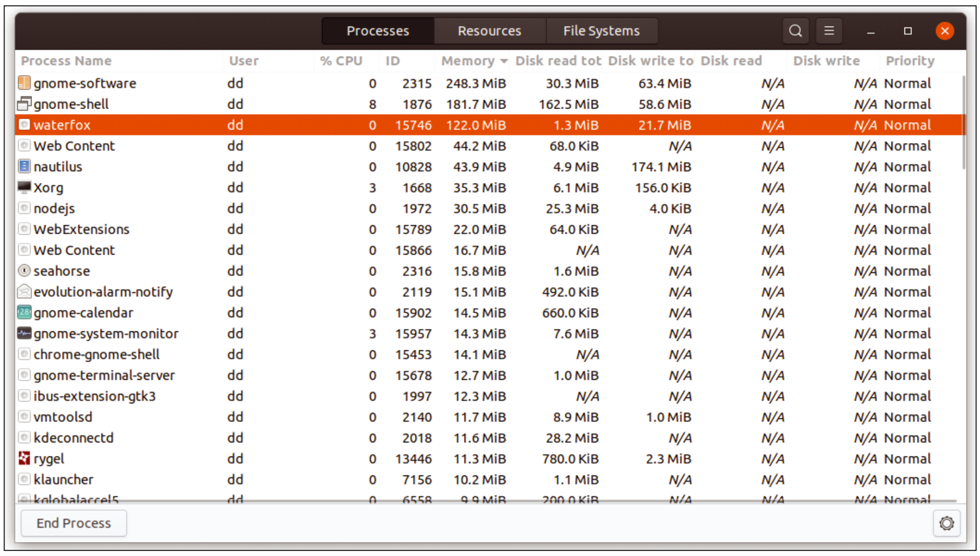
Another difference between the two versions is that the project actively develops the Current version, while the Classic browser is only given regular updates to fix bugs and plug vulnerabilities.

Both variants are available as tarballs for 64-bit architectures. They weigh in at around 58MB (Current) and 88MB (Classic). The hardware environment the developers say you need is 512MB RAM, 200MB free space on disk, and a processor with SSE3 capabilities. These SSE3 extensions are offered by almost all full-fledged Intel and AMD processors for desktop and mobile computers from Pentium 4 upward. In addition, the software requires at least version 2.28 of the GLib library.

First unpack the archive. You can move the `~/waterfox/` directory and its multiple subdirectories to a folder of your choice; `/opt/` is recommended for most distributions. In the final step, create a new entry in the menu hierarchy to start the browser. You can launch the software by typing the `./waterfox` command in the program directory.

Differences

At first sight, Waterfox looks like a complete clone of Firefox: Apart from using a different program icon, no other differences are immediately obvious. But Waterfox fires up far faster. The reason for this becomes clear when you take a look at the system load



Process Name	User	% CPU	ID	Memory	Disk read tot	Disk write to	Disk read	Disk write	Priority
gnome-software	dd	0	2315	248.3 MiB	30.3 MiB	63.4 MiB	N/A	N/A	Normal
gnome-shell	dd	8	1876	181.7 MiB	162.5 MiB	58.6 MiB	N/A	N/A	Normal
waterfox	dd	0	15746	122.0 MiB	1.3 MiB	21.7 MiB	N/A	N/A	Normal
Web Content	dd	0	15802	44.2 MiB	68.0 KiB	N/A	N/A	N/A	Normal
nautilus	dd	0	10828	43.9 MiB	4.9 MiB	174.1 MiB	N/A	N/A	Normal
Xorg	dd	3	1668	35.3 MiB	6.1 MiB	156.0 KiB	N/A	N/A	Normal
nodejs	dd	0	1972	30.5 MiB	25.3 MiB	4.0 KiB	N/A	N/A	Normal
WebExtensions	dd	0	15789	22.0 MiB	64.0 KiB	N/A	N/A	N/A	Normal
Web Content	dd	0	15866	16.7 MiB	N/A	N/A	N/A	N/A	Normal
seahorse	dd	0	2316	15.8 MiB	1.6 MiB	N/A	N/A	N/A	Normal
evolution-alarm-notify	dd	0	2119	15.1 MiB	492.0 KiB	N/A	N/A	N/A	Normal
gnome-calendar	dd	0	15902	14.5 MiB	660.0 KiB	N/A	N/A	N/A	Normal
gnome-system-monitor	dd	3	15957	14.3 MiB	7.6 MiB	N/A	N/A	N/A	Normal
chrome-gnome-shell	dd	0	15453	14.1 MiB	N/A	N/A	N/A	N/A	Normal
gnome-terminal-server	dd	0	15678	12.7 MiB	1.0 MiB	N/A	N/A	N/A	Normal
ibus-extension-gtk3	dd	0	1997	12.3 MiB	N/A	N/A	N/A	N/A	Normal
vmtoolsd	dd	0	2140	11.7 MiB	8.9 MiB	1.0 MiB	N/A	N/A	Normal
kdeconnectd	dd	0	2018	11.6 MiB	28.2 MiB	N/A	N/A	N/A	Normal
rygel	dd	0	13446	11.3 MiB	780.0 KiB	2.3 MiB	N/A	N/A	Normal
klauncher	dd	0	7156	10.2 MiB	1.1 MiB	N/A	N/A	N/A	Normal
kglobalaccel5	dd	0	6558	9.9 MiB	200.0 KiB	N/A	N/A	N/A	Normal

Figure 1: Waterfox requires far fewer resources than Firefox.

display. While Firefox consumed 250MB RAM in idle mode on our lab system, Waterfox made do with less than half that (Figure 1).

Although the two browsers look like identical twins at first glance, the Waterfox programmers, led by Alex Kontos, have completed a thorough cleanup under the hood. According to the developers, the Encrypted Media Extensions (EME), all telemetry services, sponsored tiles on new tab pages, and dubious services like Pocket have all be removed.

Waterfox also does not follow the Mozilla Corporation's annoying habit of changing interfaces, something that is particularly annoying for add-on developers. The developers have kept the XUL and XPCOM interfaces and the newer WebExtensions API. You can still use older NPAPI plugins with Waterfox. In addition, most Firefox extensions will work, once you retrieve them from the Mozilla site.

The browser integrates unsigned extensions if desired. Since the large number of different add-ons makes it difficult to keep track of the extensions, the Waterfox developers provide various archived databases on their website, from which you can source older extensions if required [2]. In addition, a special add-on named the Classic Add-ons Archive is available, which provides a database for all older extensions compatible with the alternative browser (Figure 2).

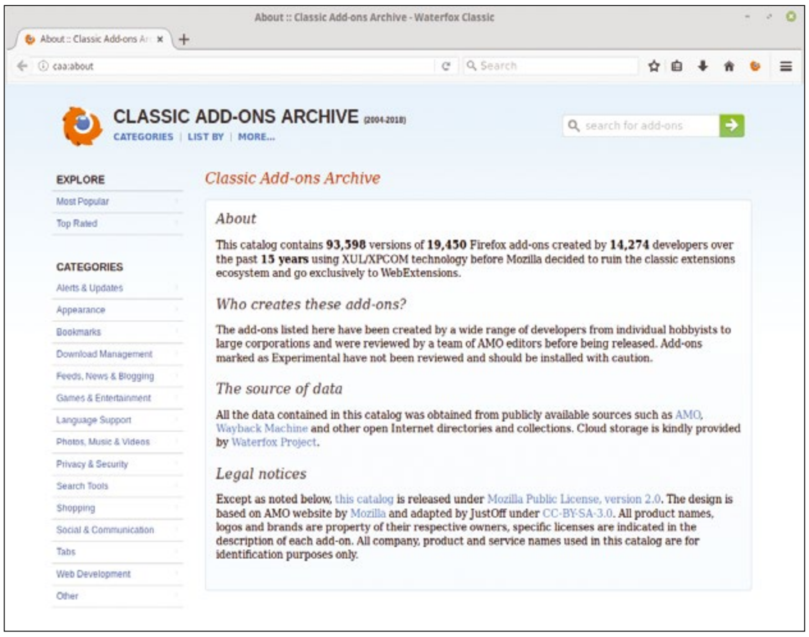
This database currently contains nearly 20,000 extensions for the browser. You can find the 43MB extension on GitHub [3]. You can use this database by clicking on the small orange button in the top right corner of the toolbar after installing the extension. Waterfox opens the archive's internal website where you use an input field to search for add-ons in the top right corner.

Alternatively, first select a group in the vertically arranged category bar on the left; the add-ons available in it then appear in the right panel of the browser window. As soon as you mouse over one of them, a green button labeled *Install Now* pops up on the right.

Clicking on it will set up the respective add-on in the browser. Extensions developed for the current version of Firefox can be installed via the corresponding website. You can also retrieve and integrate themes from there. Waterfox also lets you use older and current add-ons simultaneously (Figure 3).

Conclusions

Waterfox is an impressive example of what a better version of Firefox can look like: leaner, faster, and without the desire to collect your data. In addition, you can still use unsigned



Classic Add-ons Archive

About

This catalog contains **93,598** versions of **19,450** Firefox add-ons created by **14,274** developers over the past **15 years** using XUL/XPCOM technology before Mozilla decided to ruin the classic extensions ecosystem and go exclusively to WebExtensions.

Who creates these add-ons?

The add-ons listed here have been created by a wide range of developers from individual hobbyists to large corporations and were reviewed by a team of AMO editors before being released. Add-ons marked as Experimental have not been reviewed and should be installed with caution.

The source of data

All the data contained in this catalog was obtained from publicly available sources such as AMO, Wayback Machine and other open Internet directories and collections. Cloud storage is kindly provided by Waterfox Project.

Legal notices

Except as noted below, this catalog is released under Mozilla Public License, version 2.0. The design is based on AMO website by Mozilla and adapted by JustOff under CC-BY-SA-3.0. All product names, logos and brands are property of their respective owners, specific licenses are indicated in the description of each add-on. All company, product and service names used in this catalog are for identification purposes only.

Figure 2: The Classic Add-ons Archive gives users access to thousands of older add-ons.

and NPAPI plugins that Mozilla no longer keeps in stock. Sensible settings in terms of data protection also make the painstaking reconfiguration steps that Firefox requires largely unnecessary.

Waterfox Classic version can therefore be recommended without any restrictions for power users who also rely on older technologies. The Current version is best suited for users looking for an alternative to Firefox with faster speeds and improved privacy. ■■■

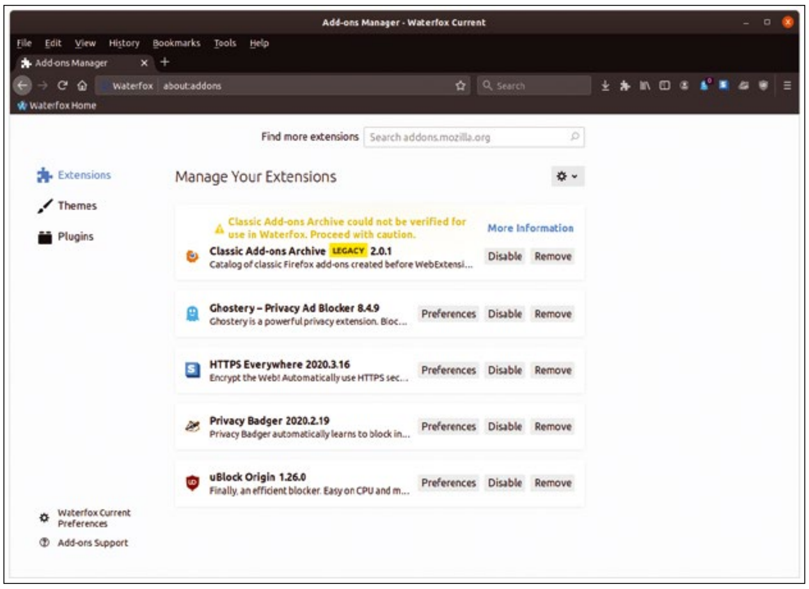
Info

[1] Waterfox: *https://www.waterfox.net*

[2] Information on add-ons: *https://www.waterfox.net/addons/*

[3] Classic Add-on Archive extension: *https://github.com/JustOff/ca-archive*

Figure 3: Waterfox supports the installation and use of older and newer add-ons in parallel.



Classic Add-ons Archive could not be verified for use in Waterfox. Proceed with caution.

Classic Add-ons Archive 2.0.1
Catalog of classic Firefox add-ons created before WebExtensi...

Ghostery – Privacy Ad Blocker 8.4.9
Ghostery is a powerful privacy extension. Bloc...

HTTPS Everywhere 2020.3.16
Encrypt the Web! Automatically use HTTPS sec...

Privacy Badger 2020.2.19
Privacy Badger automatically learns to block in...

uBlock Origin 1.26.0
Finally, an efficient blocker. Easy on CPU and m...

Using Gimp to create a simple media player skin

Picturesque

Forget the photos – Gimp is also a great tool for graphic design. This workshop shows how to use Gimp to create a simple visual image for a multimedia player application.

BY GAURAV NAWANI

Many users call on the Gimp image editor [1] to crop and touch up their digital photos, but the powerful Gimp has many other uses. For instance, developers and graphic artists use Gimp to create graphic elements for desktop applications. The details of how to tie graphics features into a working application are numerous and depend on the programming language, the desktop system, and the nature of the application. However, the steps for creating basic GUI elements in Gimp are often quite simple and serve as a useful scenario for showcasing Gimp's impressive toolbox of graphic design tools.

This workshop introduces you to some of the tools that graphic artists use to build images in Gimp. In this scenario, I'll build the skin for a sample media player application. Of course, the media player on your Linux system is a software tool that processes the contents of a stream or sound file. A media player doesn't really *look* like anything, but the metaphor of the computer desktop environment means that humans need something to look at. In this case, I'll make the media player look like an old-school CD player, with the usual buttons (Play, Stop, Previous, and Next) and a little window to show what song is playing. A more complete GUI interface would include an enlarged window for displaying playlists, as well as other elements, but for the purposes of this workshop, I'll keep it simple.

The popular Gimp comes preinstalled on many Linux systems, and if you can't find it now in the Start menu (typically with the *Graphics* applications), you can surely install it from your distro's package manager. The Gimp website also provides source code and packages for several operating systems [2]. This tutorial is based on Gimp version 2.10.14; if you are using a different version of Gimp, the details might differ slightly, but the concepts are similar.

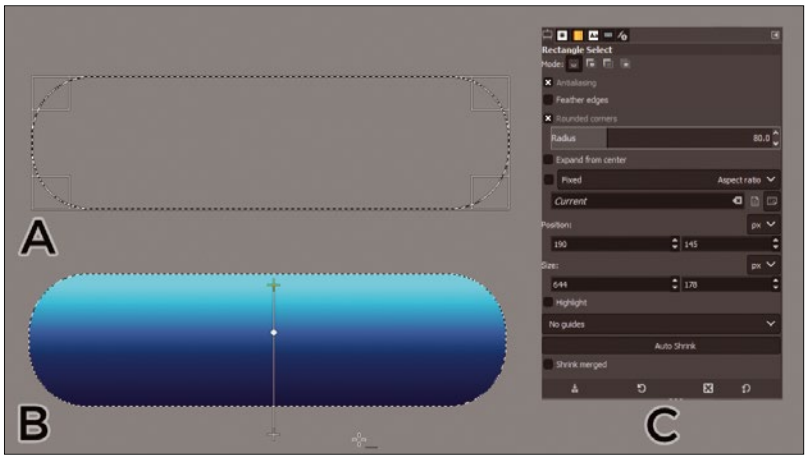
Building the Base

The first step is to create a new image. In the Gimp main window, choose the File menu and select *New*. In the Create a New Image dialog, select an image size of 1024x512 pixels.

Next, click *Layer | New Layer* to create a new transparent layer. Rename the layer *<skin-base>*.

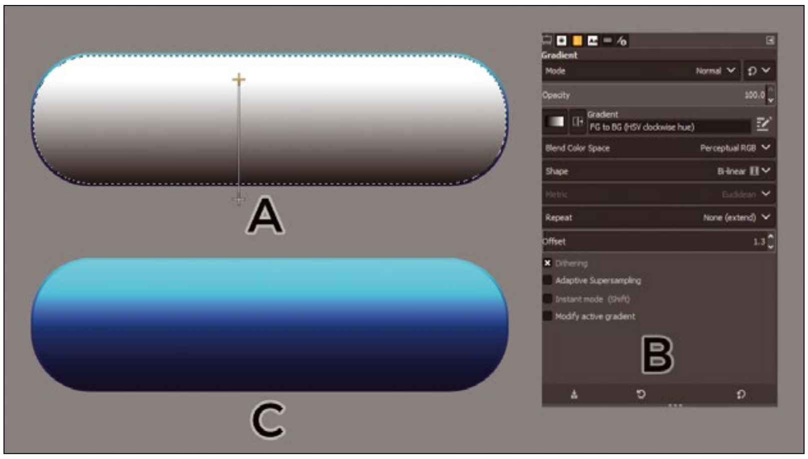
Now in the Toolbox, choose the *Rectangle Select* option and select the box labeled *Rounded Corners*. Input 80 as a radius for the rounded corners, and then draw a rounded rectangle of 644x178 pixels, as shown in Figure 1C.

The next step is to color in the rectangle. The gradient tool will give the image a polished, industrial



Rectangle Select
Mode:
[✓] Antialiasing
[ ] Feather edges
[✓] Rounded corners
Radius 80.0
[ ] Expand from center
[ ] Fixed Aspect ratio
Current
Position: px
190 145
Size: px
644 178
[ ] Highlight
No guides
Auto Shrink
[ ] Shrink merged

Figure 1: Creating a media player skin base using the selection and gradient tools.



Gradient
Mode Normal
Opacity 100.0
Gradient FG to BG (HSV clockwise hue)
Blend Color Space Perceptual RGB
Shape Bi-linear
Metric Euclidean
Repeat None (extend)
Offset 1.3
[✓] Dithering
[ ] Adaptive Supersampling
[ ] Instant mode (Shift)
[ ] Modify active gradient

Figure 2: Refining the tone and border identification of the media player skin.

look. Select the gradient tool in the toolbox, or if you don't see it, choose the Tools menu and select *Paint Tools | Gradient*. Select the preset *Deep Sea* gradient from the gradient options. Ensure the *<skin-base>* layer is selected in the *Layers* tab, and then draw the gradient as shown in Figure 1B and press the Enter key.

Keeping the selection active, now go to *Select | Shrink*. Input 3 as a value then press the OK button to confirm. Create a new transparent layer, renamed *<skin-shine>*, and use the selection to fill with a B&W gradient, where black is at the top, as shown in Figure 2. Press Shift+Ctrl+A to lose the selection.

Change the layer mode of the *<skin-shine>* layer to *Overlay* and the opacity to 50. Using the *Overlay* layer mode with the gray gradient layer adds a bit of shine to the lower layer that resonates with a metallic tint. Also, the three-pixel reduced size now acts as a responsive border for the *<skin-base>* layer. If you look carefully, you can see the difference between Figure 1B and Figure 2C.

Make the LCD panel

The next step is to create the simple LCD panel for the media player. Create a new transparent layer called *<lcd-base>* at the top of the layer stack. Then draw a rectangular selection of 400x64 pixels with a rounded radius of 16, as shown in Figures 3A and 3B. Position the rectangle in the vertical center of the *<skin-base>* layer, slightly upwards to the mid-top part. Now fill the selection with color 808f80, as shown in Figure 3A.

To add a gradient to the LCD panel, create a new transparent layer and name it *<lcd-shine>*, and then click on *Select | Shrink* menu with 3 as the value to shrink the selection. Draw a linear B&W gradient, where black is towards the top, as shown in Figures 3C and 3D.

Change the layer mode of *<lcd-shine>* to *Overlay* and its opacity to 56 (see Figure 4B). Now you have created a nice glass-screen effect over the LCD panel. You can edit the shine to enhance it. Select the *<lcd-shine>* layer, and then go to *Layers | Colors | Levels* and change *Input levels* to (56|0.44|232), as shown in Figure 4A.

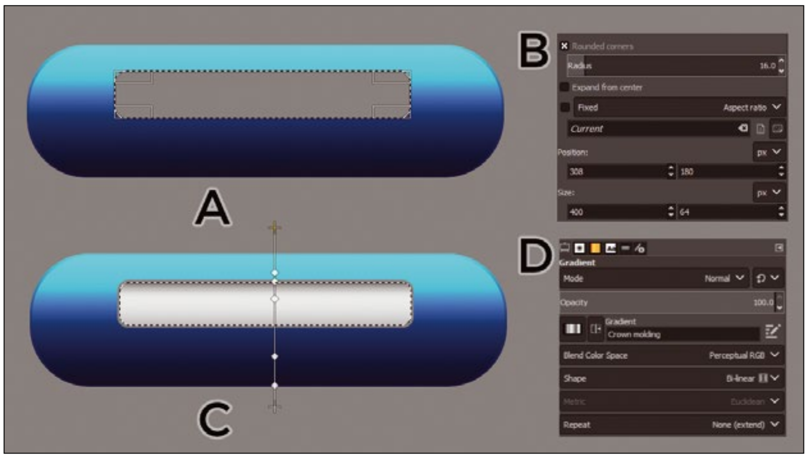
Make the Button Panel

The control buttons for the media player will appear on a button panel below the base. The first step is to create the panel; then I'll add the buttons.

Press Shift+Ctrl+A to lose the current selection. Now create a new transparent layer called *<bp-base>* on top of the layer stack. Again use the rectangular selection tool with a 420x100 pixels size and rounded corners at 30. Ensure it is aligned with the LCD panel, and keep its top half inside the *<skin-base>* layer, as shown in Figure 4C.

Keep the selection active and use the gradient fill preset *Shadows 2*, as shown in Figure 5A, and then press the Enter key to finalize the *<bp-base>* layer. Using this golden gradient distinguishes the buttons panel from the background skin.

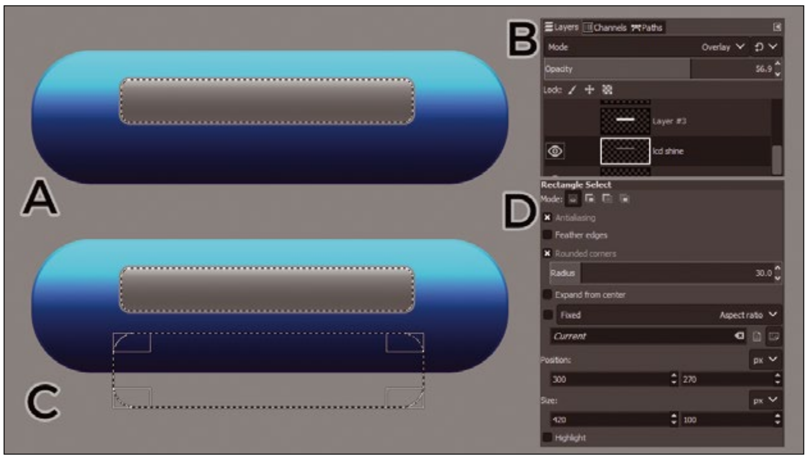
On the last active selection, do a layer shrink of 3 and create a new transparent layer named *<bp-shine>*. Again select and use the B&W gradient, as shown in Figure 5C, and then change the layer mode to *Overlay* with 30 as the layer opacity. This step refines the 3D-ness, and the gradient shine enhances the look of the button panel, as shown in Figure 6D.



B: Rounded corners [✓]; Radius 16.0; Expand from center [ ]; Fixed [ ] Aspect ratio; Current; Position: px 308 180; Size: px 400 64

D: Gradient; Mode Normal; Opacity 100.0; Gradient Crown molding; Blend Color Space Perceptual RGB; Shape Bi-linear; Metric Euclidean; Repeat None (extend)

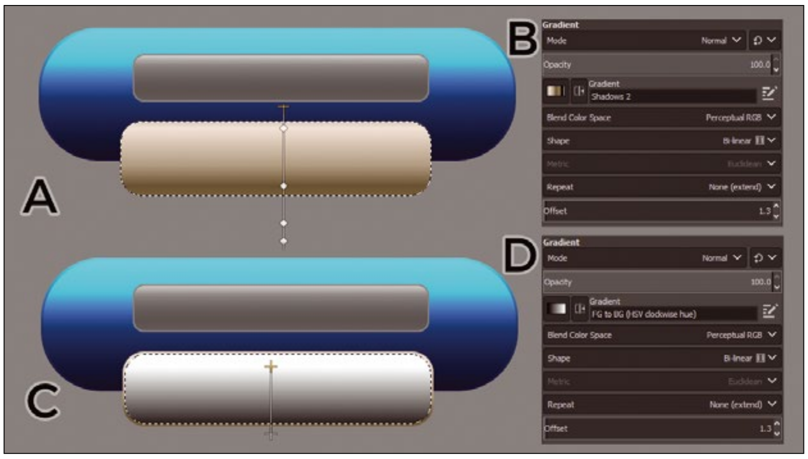
Figure 3: Creating the LCD panel.



B: Mode Overlay; Opacity 56.9; Layer #3; lcd shine

D: Rectangle Select; Mode:; Antialiasing [✓]; Feather edges [ ]; Rounded corners [✓]; Radius 30.0; Expand from center [ ]; Fixed [ ] Aspect ratio; Current; Position: px 300 270; Size: px 420 100; Highlight [ ]

Figure 4: Refining the LCD panel and creating a button panel.



B: Gradient; Mode Normal; Opacity 100.0; Gradient Shadows 2; Blend Color Space Perceptual RGB; Shape Bi-linear; Metric Euclidean; Repeat None (extend); Offset 1.3

D: Gradient; Mode Normal; Opacity 100.0; Gradient FG to BG (HSV clockwise hue); Blend Color Space Perceptual RGB; Shape Bi-linear; Metric Euclidean; Repeat None (extend); Offset 1.3

Figure 5: Refining the button panel.

The basic position of the button panel in Figure 5 looks fine, but something is missing. For aesthetics reasons, I would like to see more separation of the button panel from the rest of the media player unit, so I'll create a gray border around the button panel that separates the panel from the blue cylinder of the media player base.

To proceed, I'll need another transparent layer above the *<skin-shine>* layer called *<skin-modi>*. Right-click on the *<skin-base>* layer in the *Layers* window and select *Alpha to Selection*, as shown in Figure 6A.

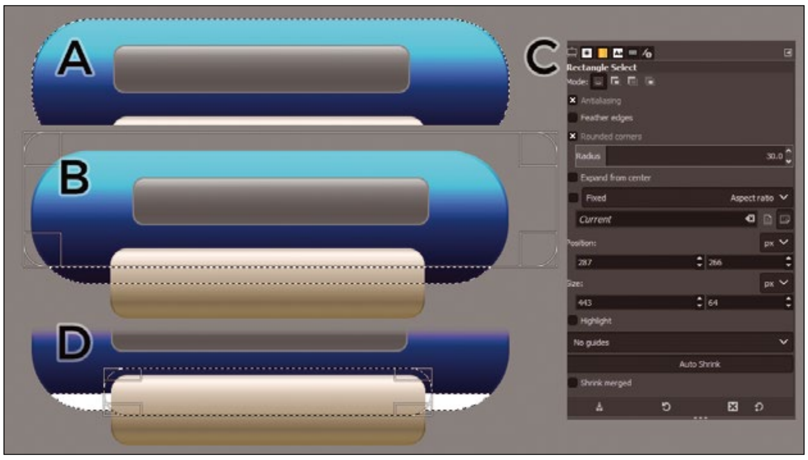


Figure 6: Using selection tools for the media player separator.

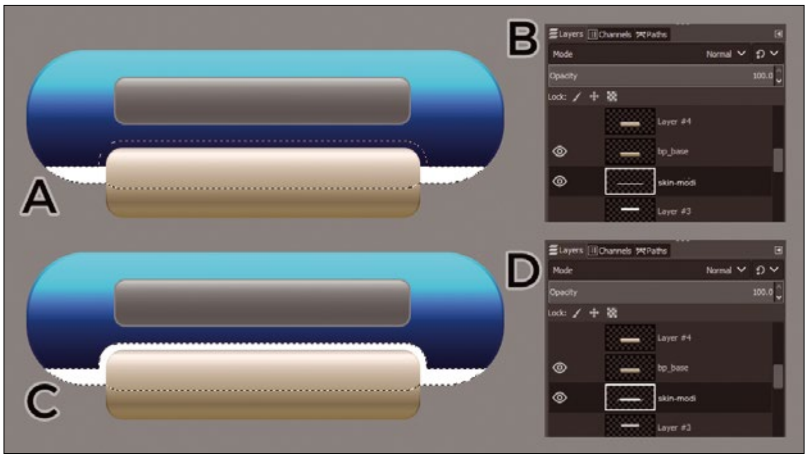


Figure 7: Giving the media player separator a refined look.

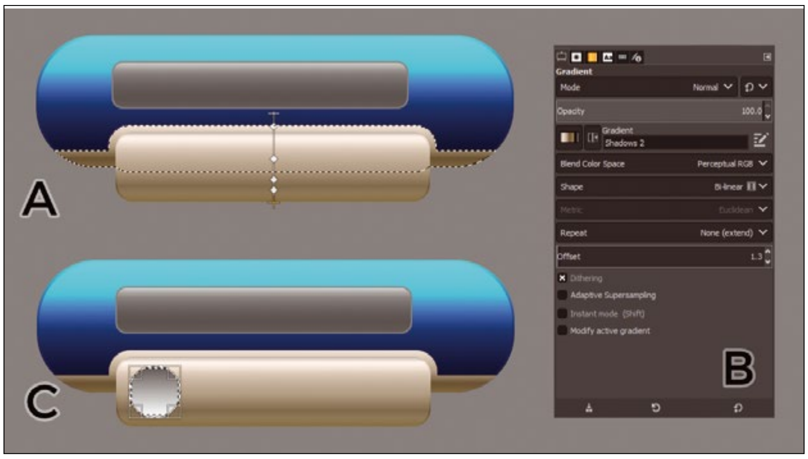


Figure 8: Finishing the panel separator's look and creating a button for the Button panel.

Now I will cut some part from this selection by keeping the Ctrl key pressed and drawing the new selection, as shown in Figure 6B.

To the selection from the previous step, I will make an addition. Keep pressing the Shift key, and draw the rounded rectangular selection, as shown in Figure 6C and 6D. Then press the Enter key to get the added selection, as shown in Figure 7A.

The next step is to fill this selection with the white color on the *<skin-Modi>* layer just to keep it safe for further use. Now hide this *<skin-modi>* layer and add another transparent layer above it called *<skin-modi-gradient>*.

Select *Alpha to Selection* from the *<skin-modi>* layer, as shown in Figure 7C. Then select the gradient preset *Shadows 2* and draw it, as shown in Figure 8A. This makes the *<skin-modi-gradient>* layer darker and serves as a good separator over the blue skin, acting as a visual cue for the buttons panel.

Making the Buttons

The last step in this Gimp mini-project is to create the control buttons on the button panel. Press Shift+Ctrl+A to lose the current selection. Now create a new transparent layer called *<button-base>* at the top of the layer stack. Click on the Tools menu and choose *Selection Tools | Ellipse Select*. Use the *Ellipse Select* tool to create a circle of 72x72 pixels. Fill the circle with the B&W gradient, as shown in Figure 8C.

Create another transparent layer called *<button-3d>* and do *Select |Shrink (3)*; then fill the button with a B&W gradient, as shown in Figure 9A. Set the layer mode to *Pin-light* with 75 as layer opacity.

The buttons are almost finished, but each still needs an icon describing the button's purpose. I'll create these simple icons by using the pixel brush to draw them on a transparent layer. Working with pixel-level graphics requires the use of a grid for a visual aid.

Create a new transparent layer called *<prev-icon>* above the *<button-3d>* layer, as shown in Figure 9B. To get the grid ready, choose *View |Show Grid* then *View |Snap to Grid*. Now go to *Image |Configure grid* to bring up the grid setting dialog box. Configure the grid spacing at 4x4 pixels and press OK. At 100 percent zoom, your image should appear, as in Figure 10A.

Zoom to 300 percent by choosing *View |Zoom | 300%*. Press the middle mouse button and drag to pan the image and bring the first button in the zoomed up view. Press *N* for the *Pencil* tool. Select *Circle brush* (4x4) from the *Brushes* tab, as shown in Figure 10B, and change the current color to 7e7e7e.

Now place the cursor over the first button, click, and drag (keep the Ctrl key pressed) for a straight line, using the bar as a reference (see Figure 10C).

The height of the icon is 32 pixels and the diagonal axis for a reverse play icon is about 22 pixels. Fill the middle part with the pencil. Also, draw the line at the left as 32 pixels in height. Now put the *<prev-icon>* layer mode to *Overlay* to make it look like the icon is embedded in the button, as shown in Figure 10C.

Duplication in Gimp is easy: Just run *Layers | Crop to Content* in each of the three buttons layers (*<button-base>*, *<button-3d>*, and *<prev-icon>*). Then choose *Layer | New Layer Group* and move these layers inside. Press Shift+Ctrl+D to duplicate this button group and move it to the right by about 80 pixels. Repeat it four times, as shown in Figure 11B.

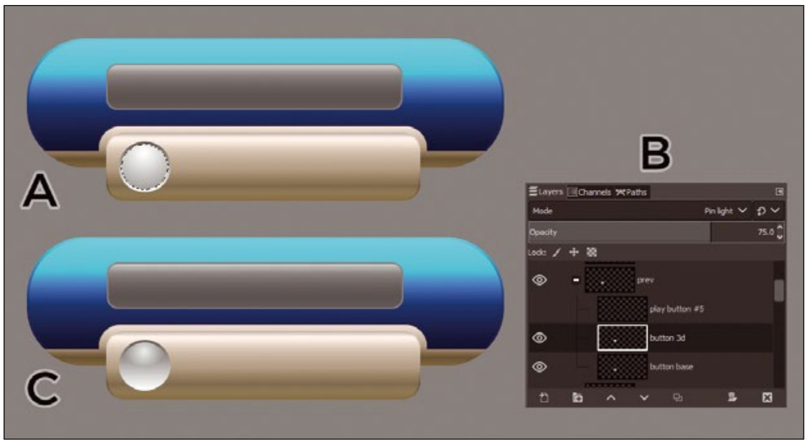
Delete the old icon layer and create a new icon layer in each button group; then draw new icons for the other buttons, as shown in Figure 12A.

The media player skin looks almost complete, but the LCD panel appears empty. If this were a real multimedia player skin, the LCD panel would receive a value from the program and display the title of the song that is currently playing. For this simple example, I'll finish by inserting some song text in an LCD-style font. Use *Overlay* for the LCD item details at about 75 to 50 percent opacity of the layers (Figure 12B).

Conclusion

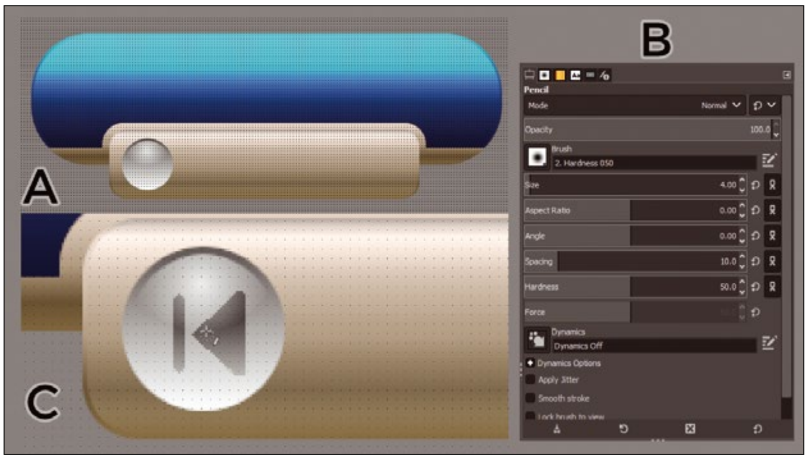
This workshop introduced you to some of the tools graphic artists use to create visual elements for desktop software. Of course, creating the image is only the beginning. In a real-world desktop tool, the graphic elements then must be incorporated into the application or installed as a skin or theme through the user interface. Several media players provide the option of changing the skin to give the player a different look – and even offer tools for inventive users to create their own custom skins. For instance, the VLC media player provides a skin editor tool [3] that allows the user to import graphic files and save them in the `.vlt` file format used for the skins in the VLC skin gallery.

Gimp is a powerful tool. Experiment and be creative. I hope exploring the techniques described in this workshop will give you some ideas that you can use in your own Gimp graphic creations. ■■■



Mode Pin light
Opacity 75.0
prev
play button #5
button 3d
button base

Figure 9: Finishing the buttons.



Mode Normal
Opacity 100.0
Brush 2. Hardness 050
Size 4.00
Aspect Ratio 0.00
Angle 0.00
Spacing 10.0
Hardness 50.0
Force
Dynamics Dynamics Off
Dynamics Options
Apply Jitter
Smooth stroke
Lock brush to view

Figure 10: Creating the Previous icon for the button.

Info

[1] Gimp: *https://www.gimp.org/*

[2] Gimp download: *https://www.gimp.org/downloads/*

[3] Creating a skin in VLC: *https://www.videolan.org/vlc/skinedhlp/basics.html*

The Author

Gaurav Nawani enjoys designing game art in all formats – be it 3D, paintings, or vectors. He is a gamer by choice, a motorbiker, a writer (*https://gauravnawani.home.blog*) and cofounder of Ironcode Gaming.

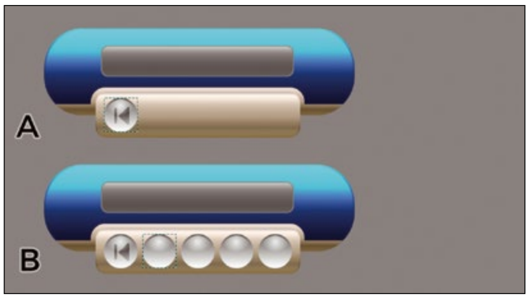


Figure 11: The Previous button finished with the icon drawn and additional buttons duplicated.

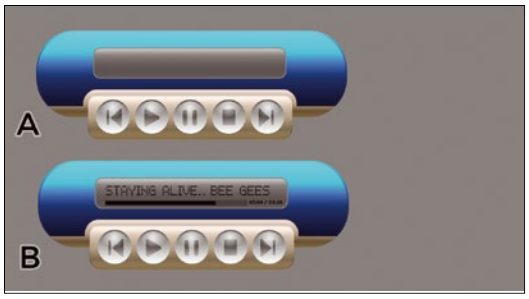


Figure 12: The final image of the media player skin.

Convert multimedia files with MystiQ

Fast Formatting

Working with multimedia files usually involves converting from one format to another. With MystiQ you can handle this task in next to no time.

BY ERIK BÄRWALDT

Multimedia content is available on the web and on data carriers in many formats. Often, your chosen media player doesn't understand all of them. This means that you will find numerous tools on Linux that convert audio or video formats. Although these tools offer a great deal of flexibility through numerous options, they are often confusing, especially to newcomers.

MystiQ [1] follows a different strategy. The program converts multimedia files, no matter whether you need to change the format of an audio file or a movie, but it keeps to the essentials. That makes it possible to convert any content with just a few mouse clicks and without extensive training. The numerous presets that you can apply to frequently used video and audio formats make the program easier to use.

Find It

The program can already be found in the repositories of several distributions, including OpenMandriva, Arch Linux, KaOS, and Slackware. You install it there with the respective package management tools.

In addition, Pling.com [2] provides packages for RPM and DEB-based distributions, and the project's GitHub site also maintains detailed installation documentation [3] for several branches of Debian. Last but not least, there is a package for Arch Linux and derivatives and a tarball with the source code. For distributions that do not support any of the precompiled formats, the project provides an AppImage.

However, all of the packages are only usable on 64-bit hardware, not on older 32-bit systems.

Try It

After installation, open the program via the starter in the menu tree. The simple input window looks pretty spartan for a multimedia application (Figure 1).

Besides a simple menubar and buttonbar, the program window is just a large area where the tool displays the contents to be converted in the form of a list. Pressing *Add files* in the top left corner lets you select the files in question. To do this, the program pops up a small file manager in an overlapping window.

MystiQ is based on the FFmpeg program collection and the associated libraries and uses their extensive functions. The program supports all common formats for files and containers.

When you select a file to convert, MystiQ transfers it to another dialog that shows you all the selected files. In this wizard, press the *Next* button, bottom right; you can then configure various settings for the output in another window (Figure 2).

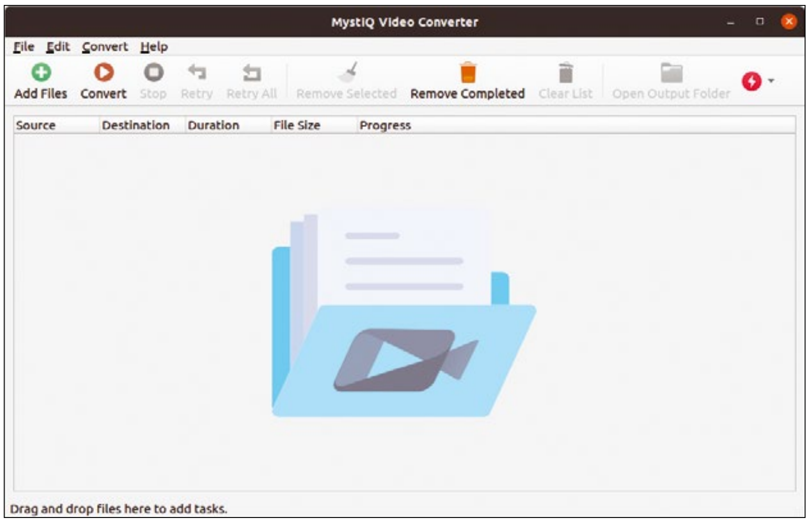

MystiQ Video Converter
File Edit Convert Help
Add Files Convert Stop Retry Retry All Remove Selected Remove Completed Clear List Open Output Folder
Source Destination Duration File Size Progress
Drag and drop files here to add tasks.

Figure 1: The MystiQ main window completely does without gimmicks.

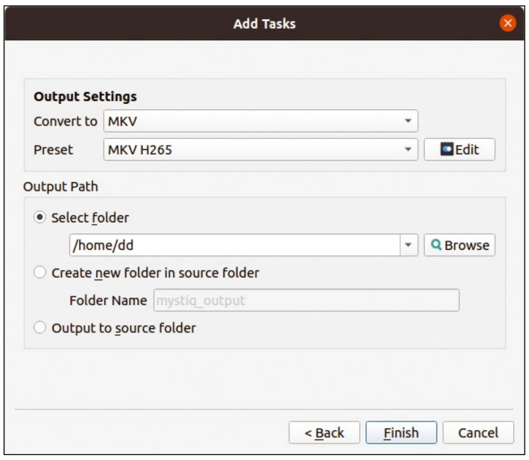


Figure 2: The wizard guides you through the program in a few steps.

Under *Output Settings*, select the desired format in the *Convert to* selection field. (For an example of choosing the format, see the box "Videos for the Raspberry Pi.") In the area below *Default Settings*, you will find options for the codec to be used. Under *Output Path*, select an existing folder or place the files in a subfolder in the source directory that the software creates.

If you want to configure more detailed options for the target file's format, press the *Edit* button to the right of the *Preset* selection field. MystiQ opens a new dialog in which you can tweak various parameters for audio and video tracks in several tabs.

Among other things, you can disable video or audio tracks, change the dimensions, or insert subtitles. There are also functions for rotating and mirroring the videos and – in a separate tab – a simple editing function (Figure 3). For audio files, most formats offer the possibility to adjust the bit rate.

Please note that the software shows all options regardless of the source file format used, but by default incompatible options are grayed.

After you have adjusted the settings, press *OK* at bottom right in the window and then press the *Finish* button at bottom right in the *Add Tasks* parent window. The software then transfers the selected files to the list in the main window.

A bar appears to the right in the *Progress* column. It initially stays at zero; you need to press *Convert*, top left, to tell MystiQ to convert the files to the target format. On modern CPUs, the program will use multiple threads if necessary, helping it to convert even large files very quickly. The progress bar is individually adjusted for each file.

Considerations

MystiQ can convert several audio and video files of different formats simultaneously in a single operation. For each file, you can make individual adjustments in the wizard and the associated dialogs, including the paths for the output.

When capturing the source files, however, make sure that the program handles files with the same attributes as a group when adding them via the *Add Tasks* option. In cases where you want to convert content to different output formats, you will only want to group files for which the same target format is intended. To view the individual settings again, mouse over the files in the list before converting. The tool shows you the important data for each file.

MystiQ can handle both lossy and lossless audio formats. It is important here to keep in mind that converting to a lossless format such as FLAC can result in very large files.

Videos for the Raspberry Pi

The conversion wizard in MystiQ lets you convert media files to many formats, but it does not support the user in choosing the right format. To prepare videos for the Raspberry Pi, for example, the MKV container format is the best choice. The first three generations of the Rasp Pi have a hardware decoder for MPEG4/H.264 encoded videos. For these Rasp Pis, you would want to use the default *MKV MPEG4* setting. The hardware of the current Raspberry Pi 4, on the other hand, is now also capable of decoding H.265/HEVC. For best possible quality in combination with small file sizes, select *MKV H265* in this case.

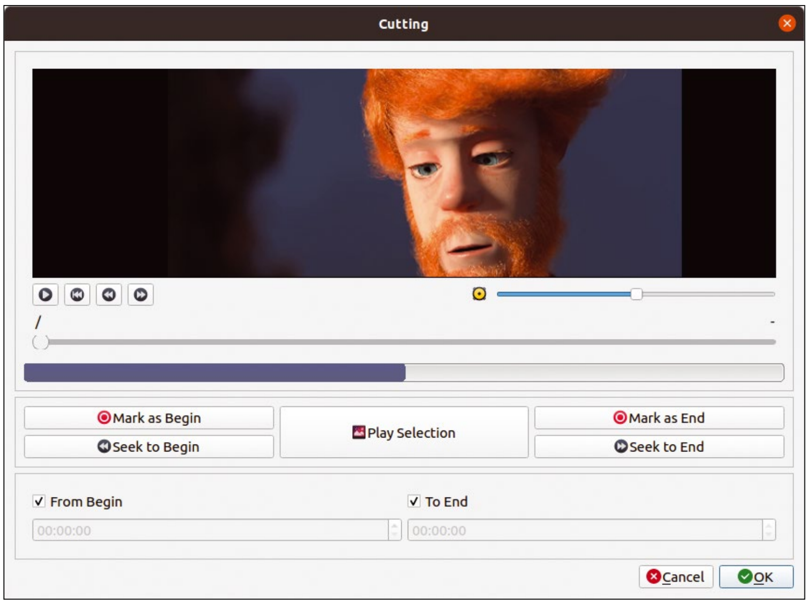


Figure 3: MystiQ even comes with a simple editing function.

Conclusions

As a well-thought-out program for converting multimedia content, MystiQ quickly produces results without extensive training. Thanks to the many presets and the intuitive configuration, it does not restrict you to a few preset parameters. For power users who often work with audio and video files of various origins, MystiQ proves to be a real help.

MystiQ turned out to be unstable on Debian "Testing" — it crashed several times in our lab. This problem did not occur on other distributions such as Arch Linux or while using an AppImage. ■■■

Info

[1] MystiQ: *https://mystiqapp.com*

[2] Download: *https://www.pling.com/p/1340589/*

[3] Installation: *https://github.com/swl-x/MystiQ/blob/master/INSTALL*

Hone your skills with special editions!

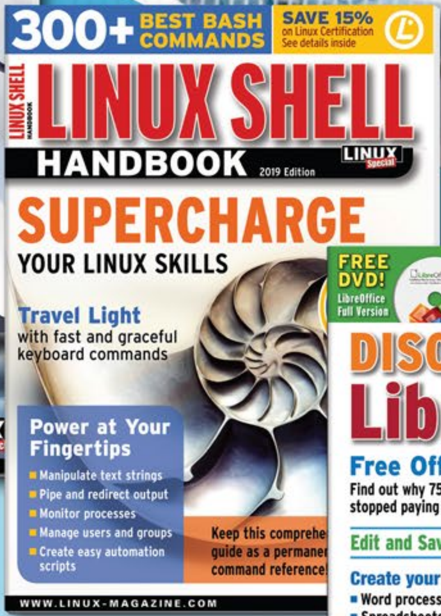
Get to know Shell, LibreOffice, Linux, and more from our Special Edition library.

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

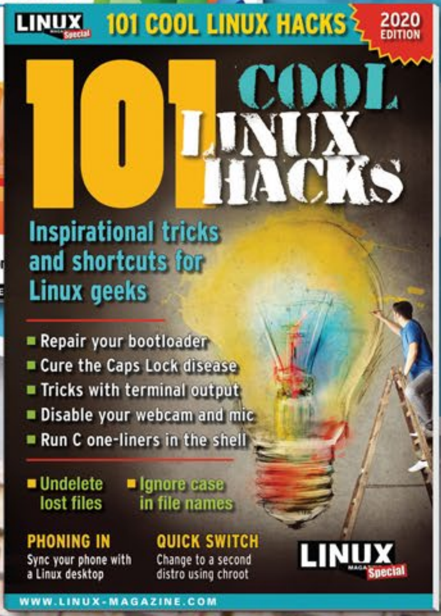
Available in print or digital format

Check out the full library!
shop.linuxnewmedia.com

FREE DVD!
ALL THE SOFTWARE YOU NEED!
openSUSE
THOUSANDS OF FREE TOOLS IN EASY REACH!
SWITCH TO LINUX NOW!
2019 Edition
GETTING STARTED WITH
LINUX
Learn how to set up a Linux system to:
Listen to Music
Surf the Web
Play Games
and Much More!
Process Photos
MORE Powerful, MORE Secure, MORE Fun!
JOIN THE LINUX REVOLUTION!
LINUX Special
WWW.LINUX-MAGAZINE.COM


300+ BEST BASH COMMANDS
SAVE 15% on Linux Certification See details inside
LINUX SHELL HANDBOOK
2019 Edition
LINUX Special
SUPERCHARGE
YOUR LINUX SKILLS
Travel Light
with fast and graceful keyboard commands
Power at Your Fingertips
Manipulate text strings
Pipe and redirect output
Monitor processes
Manage users and groups
Create easy automation scripts
Keep this comprehe guide as a permanen command reference!
WWW.LINUX-MAGAZINE.COM

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


LINUX Special
101 COOL LINUX HACKS
2020 EDITION
101 COOL LINUX HACKS
Inspirational tricks and shortcuts for Linux geeks
Repair your bootloader
Cure the Caps Lock disease
Tricks with terminal output
Disable your webcam and mic
Run C one-liners in the shell
Undelete lost files
Ignore case in file names
PHONING IN
Sync your phone with a Linux desktop
QUICK SWITCH
Change to a second distro using chroot
LINUX Special
WWW.LINUX-MAGAZINE.COM

FOSSPicks

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

The promised profusion of extra time has failed to materialize for Graham this month, leaving him with too many synth kits to build, a table littered with components, and a leaking toilet. BY GRAHAM MORRISON

Circuit design

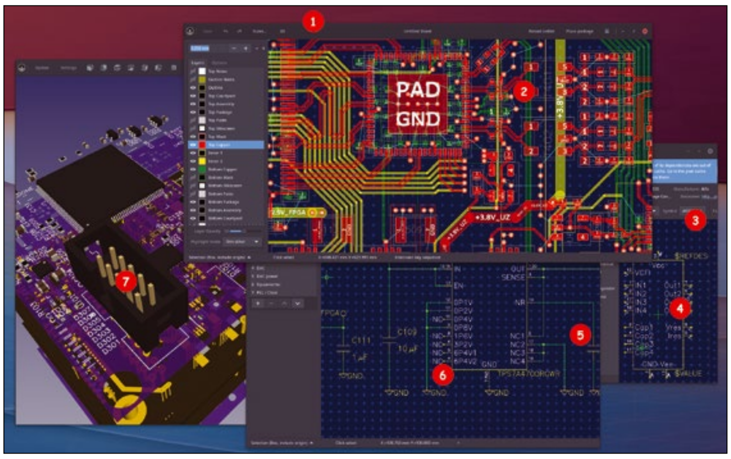
Horizon EDA

One of the many things that once seemed impossible, but is now accessible, is creating your own printed circuit boards (PCBs). You can now prototype and design your own boards from your humble home computer, and there are also now many manufacturers who welcome print runs as small as one. Add to this a crowdsourcing community happy to dive in to add their own orders and support themselves, and home brew PCB design should properly be considered a new cottage industry. The central part of this new ecosystem is open source software that can help you design, arrange, and build a circuit board that will work exactly as you intend it to and print according to your own specifications and requirements. This is the job of an Electronic Design Automation (EDA) package, which is exactly what Horizon EDA happens to be.

Horizon EDA is a comprehensive circuit design and creation tool that bundles several key components (pun intended). The first, and perhaps most important, is called "the pool." This is the parts library that lets you drag and drop predefined and readily available components into your designs. There's a comprehensive editor for adding your own parts, but you're more likely to clone the 35,000 parts database from Horizon's own GitHub repository and use this via the tabbed-by-category pool manager. There's a big difference in the way EDA implements versus other open source tools, because it uses a very flexible hierarchy of JSON-formatted files for each part, tracked by metadata stored in SQLite. It can also only contain real parts with full part numbers you can type into sites like Digikey.com to place an order. This is something you'll appreciate if you've used, for example, the Arduino IDE parts list.

With a new project created, the second key component is the schematic editor. This is where you place components from the pool to design the circuit with the functionality you need. The connections themselves are composed of networks, such as tracks for different voltages and signals. These networks can have their own algorithmically controlled rules, such as clearance, track width, and hole size. These become apparent in the third key component, the board layout preview. This is where your design becomes an actual circuit with components placed on a board. This view can quickly become complicated when it arranges your components and circuits across the various layers and tracks required to take the logical concept into the physical realm. You can even view the final board in exploded OpenGL 3 3D, which is a brilliant educational tool useful for any example boards you might want to download or share.

The GTK+ 3 interface is absolutely beautiful throughout, and the development team has succeeded in making their application a playground for experimentation. It's also brilliant having everything in one place, without having to skip between different applications or worry about whether one layer is going to be properly translated into a circuit. When you're finished, you don't just get a PCB you can order; you also get a fully populated and formatted bill of materials ready for your component order. If you've ever had to do this manually, you'll know this is half the effort of the entire circuit design. That all this can be found in a new and yet fully mature open source application is remarkable. There's never been a better time to brush up on your electrical engineering knowledge!



1. **Board layout:** After creating a circuit schematic, the board editor renders its netlist of circuits into a printable circuit board. 2. **Real parts:** Every component includes annotated parts and context-aware details on their specifications. 3. **Parts pool:** There are over 35,000 different components in the additional parts repository. 4. **Parts editor:** Add and annotate your own parts to use in schematics. 5. **Schematic editor:** This is where the logic of your circuit is designed using components from the pool. 6. **Netlists:** Individual circuits within your design are added to the netlist, which are directly translated onto the board layout. 7. **3D/2D view:** See exactly what your populated board will look like, complete with layer explosion.

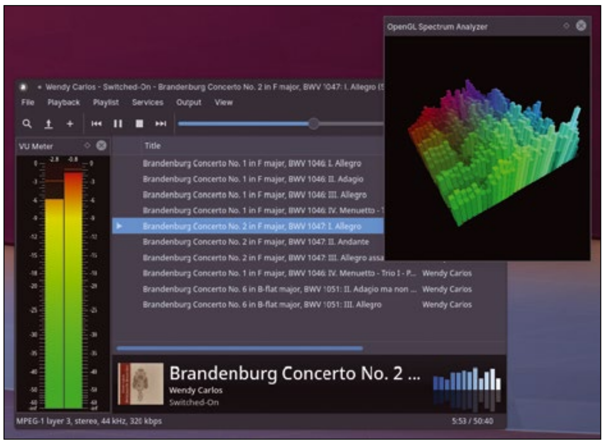
Project Website
https://horizon-eda.org

Music player

Audacious 4.0

People are passionate about the kinds of music they enjoy, which is perhaps why there are so many different music players for Linux. From the old school electro minimalism in XMMS to the audiophile prog rock madness of Amarok, there's something for every kind of listener with every kind of taste. Among this plethora is Audacious, a new favorite of ours. *New* is perhaps the wrong word for a project that's been around for 14 years, but it's the user interface (UI) overhaul in this release that's made the new version so effective. Audacious 4.0 switches its default graphical toolkit from the ancient GTK2 to the (soon to be deprecated) Qt 5, although in defense of the Qt port, moving to the imminent Qt 6 should be painless.

Moving to Qt has made a huge difference to the UI. By default, it's distraction-free and very clean. Drop the files you wish to play into the main window, and they become the play queue. Press play, and you simply have the output, complete with simple spectrogram, album thumbnail, and track description. It looks fantastic and is exactly what we'd consider an ideal UI for listening to music. But this being Qt, it's also easily augmented. Audacious provides many different UI embellishments, from a deluge of visualizers, including a 3D spectrogram and excellent VU meter, to lyrics, song details, and a list of online radio stations. When a new pane is enabled, it can be freely dragged within the main window or broken into a separate floating window. All these options are easily enabled and disabled from the excellent settings panel, from which you can also enable all kinds of audio effects if you need them. It's one of the best interfaces we've seen – simple and easy to navigate while somehow managing to package together as many options as even the most feature-rich music player.


If you don't like the new UI in Audacious, you have the option to switch it back to a good old-fashioned XMMS skin.

Project Website
https://audacious-media-player.org

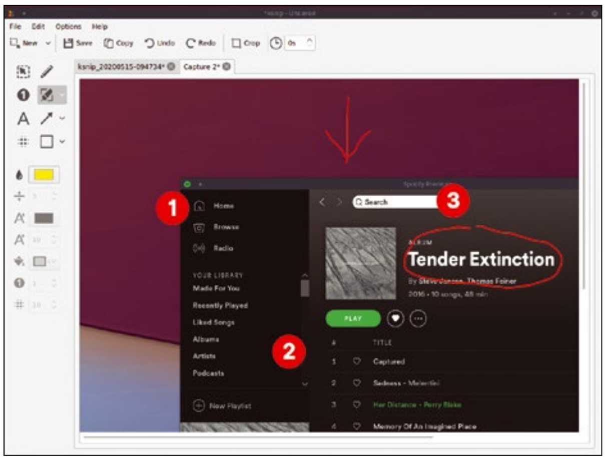
Screen grabber

ksnip

If there's one tool you rely on when writing these pages (other than a text editor, of course), it's the tool you use to make screenshots. It's not until you start taking lots of them, under all kinds of circumstances, that you realize there's a big difference between the old Windows equivalent of pressing PrtScn and hoping for the best, and something advanced enough to make your life easier. Specifically, it helps if a screenshot tool is unobtrusive, allows for a timer, saves with a sensible file name, remembers the save directory, works with OpenGL/compositing, and can grab a user-defined section of the screen. It's amazing how many screenshot tools there are, and yet how few can do all of this.

For a long time, my favorite screenshot tool was Plasma's default, Spectacle. It meets all of the above requirements and is a pleasure to use. It's particularly good at completely getting out of the way when you take a screenshot. But it was eventually beaten into second place by something called HotShots for one simple reason – HotShots incorporates an annotation tool. This is the icing on the cake when you need to leave notes and reminders on your screenshots for time-pushed magazine designers. But HotShots, frustratingly, never defaults to saving as PNG images and is way more clunky than Spectacle.

Fortunately, there's another tool that offers the best of both worlds, and it's another Plasma-based screenshot utility. Ksnip has quietly been improving for years, and it's impressive how far it has progressed. Its selection tool is the best we've encountered, with its crosshairs and updating magnified view of the cursor. It will even automatically upload images, either to Imgur or by using your own predefined script. But its best feature is the integrated editor that lets you smoothly draw, highlight, scale, crop, and numerically annotate your screenshots, saving huge amounts of time. You can even load external images and have more than one open at once. The only thing it can't do is take a screenshot of itself!



Alongside its automatically numbered annotations, ksnip also lets you open previously saved image files and annotate them in the same way.

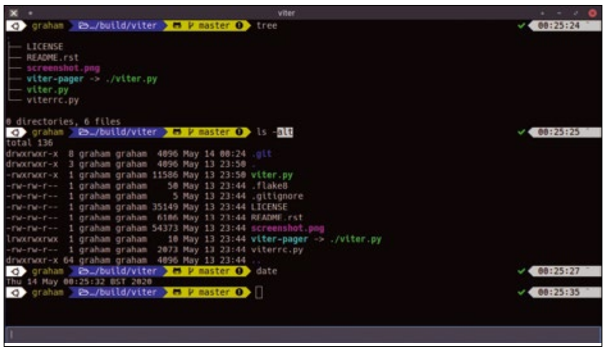
Project Website
https://github.com/ksnip

Terminal emulator

Viter

My longtime favorite terminal emulator is Konsole running ZSH. Konsole is obviously an important part of the KDE Plasma desktop and integrates perfectly with color themes and font selection. But it's also fast and functional, with native split support, tabbed views, key control over profile swapping, and an unlimited scrollback. But this doesn't mean there aren't better terminal emulators out there. Some that we've previously looked at have been seriously impressive, such as the GPU-accelerated Alacritty. But Viter could be another candidate for one simple reason: It's a terminal emulator that attempts to use Vim key bindings alongside similar *normal* edit and *detached* command modes for terminal interaction. While Bash does have a setting for Vim key bindings, and there are various other hacks you can try, none of these feels quite right, because they're not a native part of the terminal emulator's design. Vim is so native to Viter that it even lends itself to the name.

Built with Python 3, Viter is also easy to run with very few dependencies, although we did need to manually modify the `#!` part of the script to point to the Python binary at `/usr/bin` rather than `/bin`. With that done, there's little to initially tell this terminal emulator apart from another. It will use your favorite shell environment and respond to commands just as any terminal emulator should. The fun starts when you press Ctrl+Shift+Space. This is the invocation to put you into *detached* mode, which is the equivalent to *command* mode in Vim. You can now use Vim navigation to view the scrollback, including *g* and *G* to jump to its beginning and end respectively, */* to search, *n* for the next occurrence, and *y* to copy. You can then use *:* to execute a Python interpret command or *e* to evaluate a Python expression. It works well, and while maybe not mature enough to replace Konsole just yet, it's a great idea.



```
graham  ../build/viter  master  tree                                  00:25:24
.
├── LICENSE
├── README.rst
├── screenshot.png
├── viter-pager -> ./viter.py
├── viter.py
└── viterrc.py

0 directories, 6 files
graham  ../build/viter  master  ls -alt                               00:25:25
total 136
drwxrwxr-x  8 graham graham  4096 May 14 00:24 .git
drwxrwxr-x  3 graham graham  4096 May 13 23:50 .
-rwxrwxr-x  1 graham graham 11586 May 13 23:50 viter.py
-rw-rw-r--  1 graham graham    50 May 13 23:44 .flake8
-rw-rw-r--  1 graham graham     5 May 13 23:44 .gitignore
-rw-rw-r--  1 graham graham 35149 May 13 23:44 LICENSE
-rw-rw-r--  1 graham graham  6106 May 13 23:44 README.rst
-rw-rw-r--  1 graham graham 54373 May 13 23:44 screenshot.png
lrwxrwxrwx  1 graham graham    10 May 13 23:44 viter-pager -> ./viter.py
-rw-rw-r--  1 graham graham  2073 May 13 23:44 viterrc.py
drwxrwxr-x 64 graham graham  4096 May 13 23:44 ..
graham  ../build/viter  master  date                                  00:25:27
Thu 14 May 00:25:32 BST 2020
graham  ../build/viter  master                                        00:25:35
```

With Vim key bindings, an edit and command mode, and a Python interpreter, Viter is an excellent new take on terminal emulation.

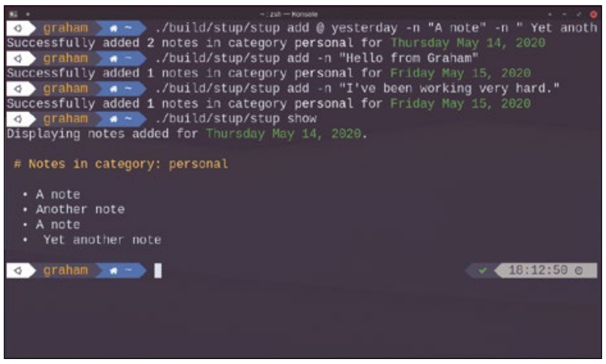
Project Website
https://github.com/Kharacternyk/viter

Stand-up notes

stup

In our day jobs, many of us need to spend a certain amount of time each and every day telling our colleagues what we've been working on. This much loved/hated/derided ceremony is known as a "stand-up," because in the olden days when developers were working in an office together they would typically all stand up to deliver their summaries. Based on the principles of agile software development, it was meant to be an opportunity to briefly summarize what you'd done, what you're doing, and whether anything was blocking you. Standing helped to keep the meeting short. These meetings have since become an important part of professional daily life, especially now with the huge increase in the number of people working from home who have to validate their existence. If you've ever been part of this stand-up process, you'll know that unless you make notes of the things you've been working on, it becomes incredibly easy to simply forget everything you've done. The result is that you end up looking like you've spent your time lounging around at home watching YouTube.

Enter stup to save your job. This discovery is a simple command-line application that takes the pain out of remembering what you've been working on by helping you make your own notes in a text file. Typing `stup add -n "this is a note"` will add a comment on what you're currently working on. You can also insert `today` or `yesterday` to assign the comment to a specific day, which is useful if your stand-up is halfway through the day. You can even add notes for the future or for a specific date. You can then retrieve your notes with the `show` argument, which defaults to the notes for the current day. This is exactly what you need when you realize you're already two minutes late and quickly need some visual notes before your brain blanks. When it comes to a monthly or annual review, you can then use the `log` argument to list all the things you've worked on within a specific date range. It's simple, but it works extremely well.



```
graham  ~  ./build/stup/stup add @ yesterday -n "A note" -n " Yet anoth
Successfully added 2 notes in category personal for Thursday May 14, 2020
graham  ~  ./build/stup/stup add -n "Hello from Graham"
Successfully added 1 notes in category personal for Friday May 15, 2020
graham  ~  ./build/stup/stup add -n "I've been working very hard."
Successfully added 1 notes in category personal for Friday May 15, 2020
graham  ~  ./build/stup/stup show
Displaying notes added for Thursday May 14, 2020.

 # Notes in category: personal

  • A note
  • Another note
  • A note
  •  Yet another note

graham  ~                                                      18:12:50
```

Never forget what you've been working on with stup on the command line.

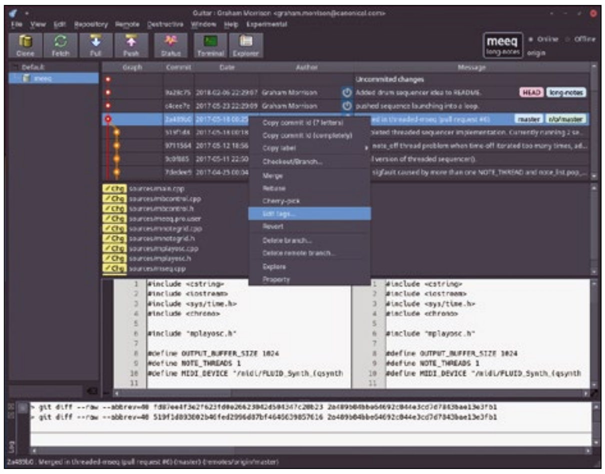
Project Website
https://github.com/iridakos/stup

GUI Git client

Guitar

As has been written many times in these pages, the version control tool, Git, has become almost as transformative within the computing industry as Linux itself. That they're both the product of Linus Torvalds' brain is remarkable, but it might explain why they can both be prone to logarithmic complexity. On the surface, Git is relatively straightforward to use. The commands to create a new branch, stage modified files, commit changes, and push those changes back are easy to understand and will mostly be all you need. But as soon as a project reaches a certain level of maturity, when the commits start conflicting and the blame log starts growing, the complexity becomes corporeal. This is where a GUI can help, both by enforcing a set process to the way you work with Git, and by helping to visualize the changes you're making to your local branches and the remote branches of your project's origin.

Guitar is a new GUI client for Git that can really help you visualize all these changes. After you've set up the paths to the `git` and `file` binaries and opened a local Git repository, the main view lists all the commits made to that repository. Each commit has easy to understand color coding and annotation, along with the widely used pipe graph to show diverging branches and commits. Select one of these commits and the *diff* view showing the file differences is immediately displayed beneath the commit log. Even more useful for new users is that the Git command used to perform any operation within the GUI, from cherry picking to rebasing, is shown in the pane beneath the diff. It's a brilliant way to use Git and to learn more about how it works at the same time.



```
#include <cstring>
#include <iostream>
#include <sys/time.h>
#include <chrono>

#include "mplayosc.h"

#define OUTPUT_BUFFER_SIZE 1024
#define NOTE_THREADS 1
#define MIDI_DEVICE "/midi/FLUID_Synth_(qsynth
```

```
> git diff --raw --abbrev=40 fd87ee4f3e2f6231d0e26623042d504347c20b23 2a489b04bbe54692c044e3cd7d7843bae13e3fb1
> git diff --raw --abbrev=40 519f1d893802b46fed2996d87bf4640639857616 2a489b04bbe54692c044e3cd7d7843bae13e3fb1
```

2a489b0 : Merged in threaded-mseq (pull request #0) (master) (remotes/origin/master)

Despite so many people using Git, there aren't that many open source GUI clients that can replace the command line. Guitar gets close.

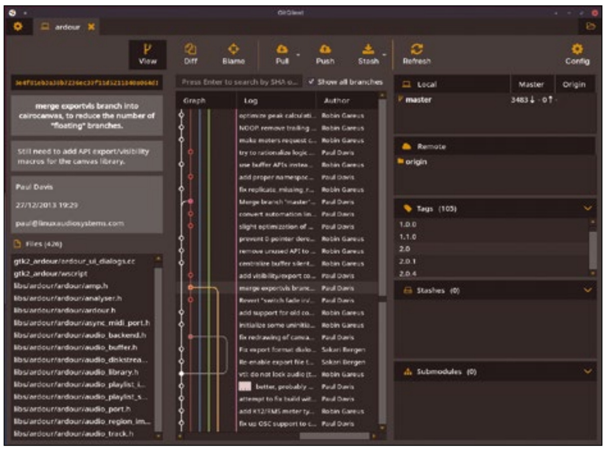
Project Website
https://github.com/soramimi/Guitar

GUI Git client

GitQlient

If you're looking for a slightly more ambitious Git GUI than the excellent Guitar (above), GitQlient can be a great option. In particular, it's considerably faster, especially when dealing with larger projects, and can open more than one project at a time. We tested this with a local Git clone of the Ardour DAW repository, which is a large project with thousands of commits over its 15 year history. Parsing this history took Guitar around 60 seconds on our system, while GitQlient took less than 5 seconds. To be fair, GitQlient appears to only load enough commits to populate the list view until you scroll down (as most apps should), but it shows that it's a potentially more ambitious tool developed to be used with larger projects.

The ambition in GitQlient is also evident in one of its best new features – the ability to have multiple repositories in the same view. This could be useful when working with a fork, for example, or porting one set of features across to another project. The main view shows the commit graph, much like Guitar, but you need to double-click a commit to see the diff. Disappointingly, the diff is presented as the colored raw diff data rather than a side-by-side view of the original and changed state. However, you can have more than one commit diff open at once, making them easy to switch between. The main view also shows local and remote branches, tags, submodules, and even stashes.



GitQlient is a fork of QGit with the old code and UI overhauled.

The first entry (pre-commit) also displays untracked, staged, and unstaged files. These are going to be of more use in a mature project, and it's a great way of diving back into a project or understanding its recent progress. When you do commit and push code, you can choose to enable auto code formatting and easily amend a commit after it has been made.

Project Website
https://francescmm.github.io/GitQlient

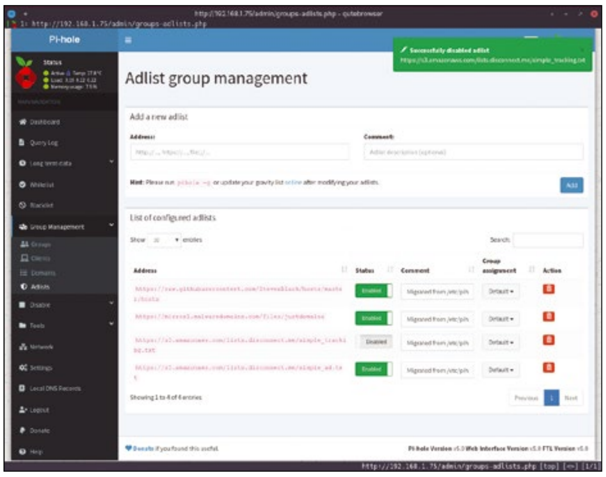
DNS ad-blocker

Pi-hole 5

All of us who work in publishing know that advertising is an essential part of the business model. Without advertising, much of what we all enjoy writing or reading simply couldn't be financed. At its best, such as in this magazine or on our website, content creators and advertisers work together to the benefit of everyone. The quality subject matter drives a specific set of advertiser needs. But at its worst, the hyper-competitive arms race between advertisers and ad-cynical content consumers has turned parts of the Internet into a tracking, privacy, and bloatware combat zone, where the only probable outcome seems to be mutually assured destruction.

The answer for many is installing an ad-blocker, which takes out both the good and the bad. They're typically installed per device or per app and are often opaque to their users. A much better solution is to install something on your network that can control all your devices and allow you much more open and transparent control over what you wish to allow through and what you want blocked. This is what Pi hole does, and it has become hugely popular. It works by becoming your local DNS, the server which converts the text web addresses you ask your browser for into the raw IP addresses that can be navigated by your device's Internet Protocol stack. But because Pi-hole controls which addresses are allowed through, it can block out those littered with trackers or malware. That removes unwanted bloat from your browsing and improves browsing speeds due to some aggressive caching, the lower bandwidth, and the lower cookie processing overhead.

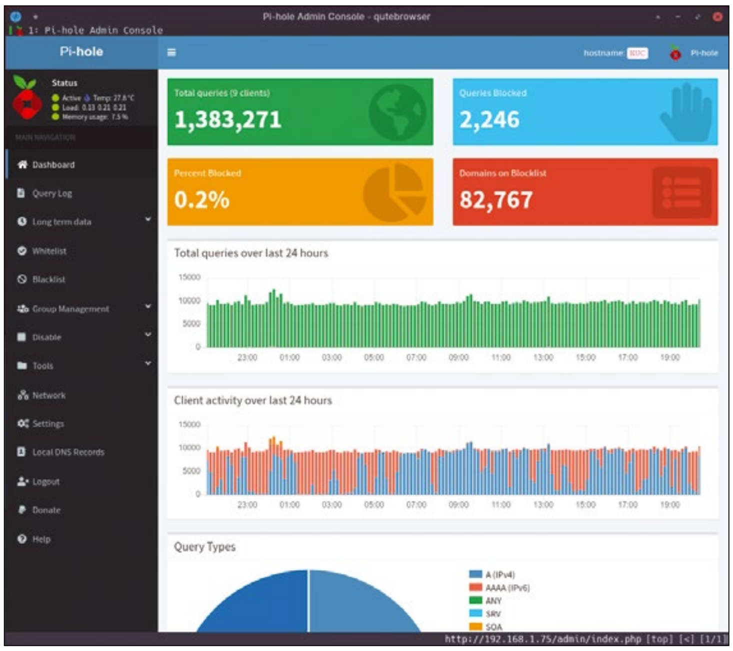
The best thing about Pi-hole is that it allows you to easily configure device access for your entire network from a single place. This means when you configure your router to use Pi-hole as its DNS, all devices on your network should then automatically ask Pi-hole first and consequently be safely filtered from the wilds of the Internet automatically. If this isn't possible, you simply need to configure each device individually to use your DNS first.



One handy use of the group management feature is to group together all devices used by children so that you can restrict their ad content more aggressively.

Another advantage of setting each device automatically is that version 5 lets you organize your devices into groups which can then be reconfigured en masse. It's incredibly useful if you want to create groups for devices belonging to children so that their content can be managed more aggressively or even disabled completely if required. This is a premium proprietary feature in many routers and services, such as Disney Circle. However, it's currently only possible when configuring each device to use Pi-hole's DNS directly, rather than configuring all devices to use a router to forward DNS requests to Pi-hole. You also can't yet set time limits to filter to a group or filters on devices coming through the router, but these are things that can hopefully be added.

It's also worth mentioning that while Pi-hole has been designed to run on a Raspberry Pi tucked away in a cupboard somewhere, it's also perfectly at home running on almost any other server you have on your network, from NAS boxes to full-fledged Linux machines. Installation is usually simple with a single script performing nearly all the configuration, dependency management, and automatic updates. It really is one of the best things you can install on your network.


Total queries (9 clients) 1,383,271
Queries Blocked 2,246
Percent Blocked 0.2%
Domains on Blocklist 82,767

The brilliant web interface lets you see just how much ad and tracking traffic is passing through your network – it's sometimes as high as 30 percent!

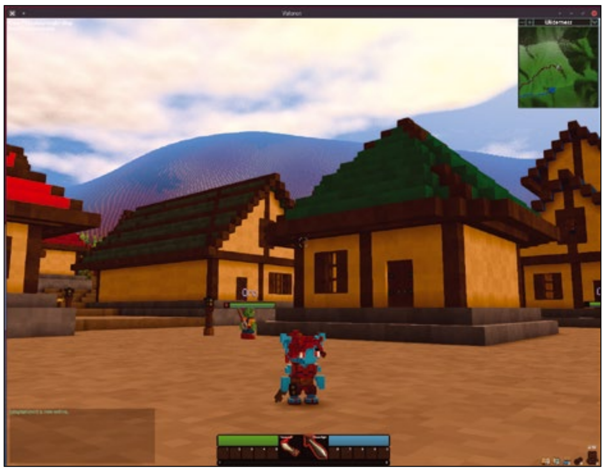
Project Website
https://pi-hole.net/

Voxel MMORPG

Veloren

Veloren is the beginnings of a beautiful, open world, open source, voxel-rendered, massively multiplayer online role-playing game (MMORPG). It's a game that was initially inspired by another game, Cube World, during the latter's seemingly moribund development. But Veloren has been able to generate enough momentum of its own to move beyond its original inspiration (and escape the negativity surrounding Cube World's early beta release) to become what looks like an incredibly promising title. You can't help but notice that the graphics have taken a huge chunk of inspiration from Minecraft, especially if Minecraft is the only voxel-rendered game you know. But Veloren uses this graphical style simply as a toolkit for its game mechanic, because it allows for huge procedurally generated worlds that can be easily generated and rendered. It's also written in Rust. Despite the size of its landscapes and the quality of its graphics, it takes very few system resources.

Though this world's development is still in the Proterozoic stage, recent releases have added procedural towns, fields with crops, and villagers with tools and clothing. There are paths and bridges between towns, simple dungeons, and saved character progression. When you generate a new game, or join one being run on someone else's machines, you can already interact with these elements and other players and even engage in simple combat. There's a map, spell system, and even archery! While you can't currently engage in the game like a full-fledged release, a lot of fun can be had exploring the vast and randomly generated terrains, walking across mountains and through forests, and discovering new towns and any number of wonderfully designed flora and fauna. All of this, including the enchanting music, has been created by the Veloren community and released as open source.


While at first glance, the graphics look like Minecraft, there's much more detail in the characters, animals, and other objects.

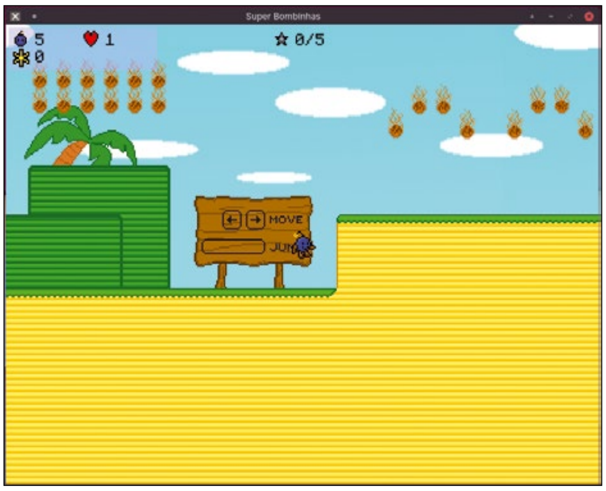
Project Website
https://veloren.net/

Classic platformer

Super Bombinhas

Despite the humongous processing and graphics advances made in the last 35 years, and the fact that the genre is still popular, the humble platform game still has the same basic mechanics as 1983's Manic Miner. You move left and right and jump to negotiate a path across one platform to another while avoiding enemies and pitfalls. Whether it's playing Moss in virtual reality, or New Super Mario Bros. U Deluxe on the Nintendo Switch, the vast majority of platform games use this same premise. Which is why it's a totally legitimate endeavor for Super Bombinhas to unashamedly pursue that old addictive gameplay style in the 21st century. Pursue is the operative word here, because the game is far from complete. There are currently six playable stages, complete with artwork, sound, and music. You can play these stages properly, collecting coins and navigating across the retro colorful 2D terrain much like you would on an 8-bit console.

The game obviously takes a lot of inspiration from early Mario games and starts off with an island map for you to track your progress as you move from one level to another. A larger map implies more islands will open as the game evolves, but much of the hard work has already been done, and the moving mechanics themselves feel complete. This is often difficult to get right, but the developer has captured the addictive feel and control of these earlier games, along with the aesthetics and level design. The game itself is written in Ruby with some help from the Gosu and MiniGL libraries. The source code looks extremely well-organized, with a clear distinction between the logic and level design, for instance. This should make it an ideal project to contribute to, even if it's just asking for access to the level editor to help populate some of those later islands.



If you've ever wanted to learn Ruby without building a web framework, Super Bombinhas is the perfect project.

Project Website
https://github.com/victords/super-bombinhas

Detecting collisions in LÖVE games

Bang! Ding! Crash!

To create an action-packed game with LÖVE, these are a few last things you should learn how to do – overlay fancy images to "physical" objects, detect collisions, and get input from the keyboard or mouse.

BY PAUL BROWN

LÖVE [1] is a Lua [2] framework that lets you develop fun 2D games relatively easily. We started talking about LÖVE and how to animate sprites in *Linux Magazine*, issue 234 [3], and went on to examine how to use the 2D physics engine to make things fall and bounce in issue 235 [4].

In this installment, you will see how to overlay PNG images to physical objects, check when they collide, and get input from the players. With these three things, you will be in a great place to start making your own games.

Let's get started!

Physical Overlay

In the prior article [4], you saw how you can use geometric figures like rectangles, circles, and polygons as bodies you can throw around or bounce off of. But you will reach a point where you will want to use something more visually appealing than just flat objects.

As you will remember from the article in the first installment of this series [3], a LÖVE game has three essential parts: *load*, *update*, and *draw*. The trick is that when you get to the draw bit you use a PNG image and overlay it on the physical object (which remains invisible) and use the physical object's position and rotation on the PNG.

For our example, let's use the mine image shown in Figure 1 and a circle as the physical object for the underlying body. Listing 1 shows what the object would look like written in LÖVE.

You create a Lua table object on line 1 and then define the object itself in the `init ()` function (lines 2 to 9). There are two differences with how you defined objects in the examples in the article in issue 235 [4]: The first is that you add a new attribute, `self.image`. This contains the path and name of the PNG image you will overlay on the physical object (line 3). The other is that instead of a rectangle, this time the physical object is a circle with the same radius as the PNG image of the mine, 14 pixels (line 5).

`Mine`'s `draw ()` function is also different. You are not drawing a circle, but using the circle's position and rotation to establish the PNG image's position and rotation in the play area. LÖVE's `body:getX ()` and `body:getY ()` functions supply the circle's position. The `body:getAngle ()` function provides its rotation. Applying these values to the image's `draw` function, you can place your PNG mine in space and have it spin like a physical object.

There is one thing to look out for though: The physical circle rotates

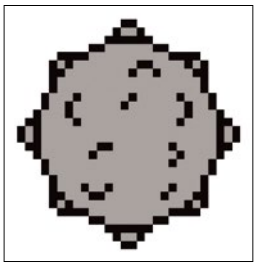


Figure 1: A PNG image of a mine you can overlay on a circular physical object.

Listing 1: pobject.lua (1)

```
01 Mine = {}
02 function Mine:init (posx, posy, bounciness, friction)
03   self.image = love.graphics.newImage ("Sprites/mine.png")
04   self.body = love.physics.newBody (world, posx, posy, 'dynamic')
05   self.shape = love.physics.newCircleShape (14)
06   self.fixture = love.physics.newFixture (self.body, self.shape, 1)
07   self.fixture:setRestitution (bounciness)
08   self.fixture:setFriction (friction)
09 end
10
11 function Mine:draw ()
12   love.graphics.draw (self.image, self.body:getX(), self.body:getY(),
                         self.body:getAngle(), 1, 1, 14, 14)
13 end
```

around its center, while the image of the mine rotates around its own origin of coordinates, which, by default, is located in the upper left-hand corner of the image. If you don't take that into account, the mine's rotation will be off-kilter. In Figure 2, you can see the image of the mine mid-rotation and the physical object (the white circle) and how they are not in sync.

You can solve this by using the offset parameters you can pass to `love.graphics.draw ()`. Instead of just writing line 12 like this:

```
love.graphics.draw (self.image, ⮐
  self.body:getX(), self.body:getY(), ⮐
  self.body:getAngle ())
```

you have to include the offset parameters and place the origin of coordinates of the image at position `14, 14` so it coincides with the center of the physical circle, as shown below.

Listing 2: main.lua (1)

```
01 require "scenery"
02 require "pworld"
03 require "pobject"
04
05 w_width = 900
06 w_height = 600
07
08 function love.load ()
09   love.window.setMode
       (w_width, w_height, {resizable = false})
10   love.graphics.setBackgroundColor
       (0.5, 0.8, 1, 1)
11
12   terrainG = Scenery
13   terrainG:init ()
14
15   terrainP = Earth
16   terrainP:init (terrainG)
17
18   mine = Mine
19   mine:init (470, 28, 0.8, 1)
20 end
21
22 function love.update (dt)
23   world:update(dt)
24 end
25
26 function love.draw ()
27   love.graphics.setColor(1, 1, 1, 1);
28   mine:draw ()
29
30   terrainG:draw ()
31 end
```

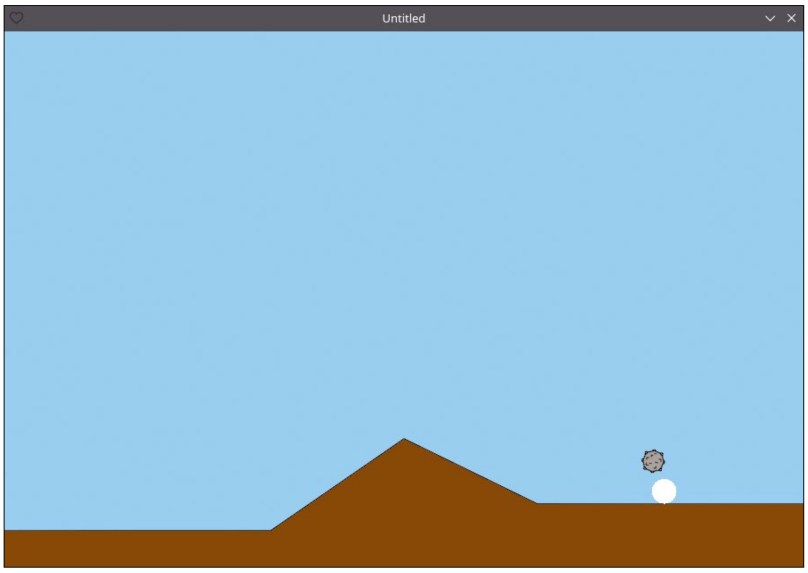


Figure 2: The different points around which the image and the physical body (the white circle) rotate prevent them from lining up.

```
love.graphics.draw (self.image, ⮐
  self.body:getX(), self.body:getY(), ⮐
  self.body:getAngle (), 1, 1, 14, 14)
```

Note that Lua does not allow for named parameters, so you also have to include the scale parameters (`1, 1`) that go before the offset parameters (`14, 14`) and that `love.graphics.draw ()` [5] requires.

The `load` function on lines 8 to 20 in Listing 2 is very similar to what you saw in *Linux Magazine*, issue 235: You create the object on line 18 and initialize it on line 19. In this case, you will be dropping your mine from position `470, 28`. As the world gets updated on line 23, the mine will fall and bounce around until it comes to a rest or bounces outside the screen (Figure 3).

Since you saw how to create the ground and world in issue 235 [4], I won't repeat that here. You

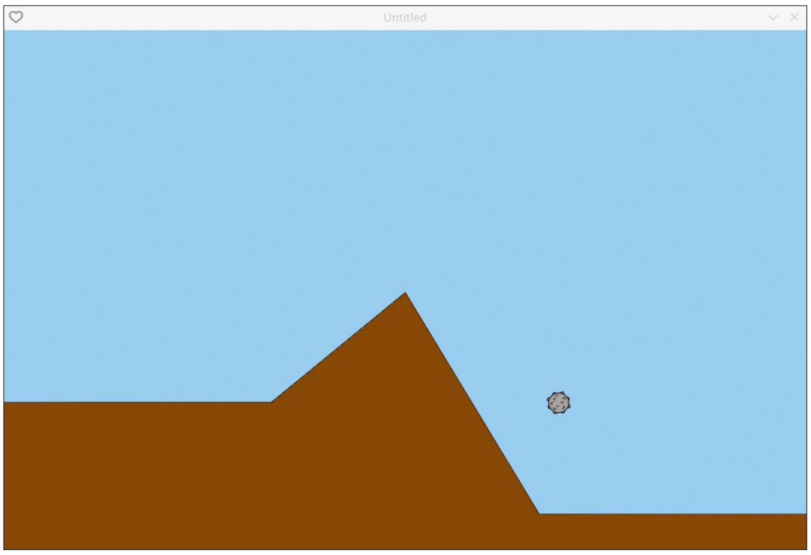


Figure 3: The mine falls, spins, and bounces around thanks to the data retrieved from the invisible physical object.

can see the complete code and not just the fragments shown here in the program's repository [6].

Colliding Objects

Once you have your mine bouncing around, you will want it to collide with something and destroy it. I drew a tank (Figure 4) and made it a body (lines 16 to 23 in Listing 3).

In this example, the mine will drop from the sky, bounce, and – hopefully – hit the tank. If that happens, the program will exit. If not, well… you'll just have to exit the program yourself.

To detect if one object has collided with another, you may be tempted to go old school and look at the position of the bounding boxes (the invisible boxes surrounding each object) and see if they are intersecting.

Don't do that.

LÖVE has Contact objects [7], which are objects that pop into existence when two bodies start to collide. You can use Contact objects to find out if objects are touching, to set the friction between colliding objects, to see how they will bounce off each other, and so on.

Figure 4: A tank we are going to drop a mine on.

Listing 3: pobject.lua (2)

```
01 Mine = {}
02 function Mine:init (posx, posy, bounciness, friction)
03   self.image = love.graphics.newImage ("Sprites/mine.png")
04   self.body = love.physics.newBody (world, posx, posy, 'dynamic')
05   self.shape = love.physics.newCircleShape (14)
06   self.fixture = love.physics.newFixture (self.body, self.shape, 1)
07   self.fixture:setRestitution (bounciness)
08   self.fixture:setFriction (friction)
09   self.fixture:setUserData ("Mine")
10 end
11
12 function Mine:draw ()
13   love.graphics.draw (self.image, self.body:getX(), self.body:getY(),
                         self.body:getAngle(), 1, 1, 14, 14)
14 end
15 ---
16 Tank = {}
17 function Tank:init (posx, posy)
18   self.image = love.graphics.newImage ("Sprites/tank.png")
19   self.body = love.physics.newBody (world, posx, posy, 'dynamic')
20   self.shape = love.physics.newRectangleShape (32, 19)
21   self.fixture = love.physics.newFixture (self.body, self.shape, 1)
22   self.fixture:setUserData ("Tank")
23 end
24
25 function Tank:draw ()
26   love.graphics.draw (self.image, self.body:getX(), self.body:getY(),
                         0, 1, 1, 16, 10)
27 end
```

But, even better, you don't have to worry about any of that, at least not for this experiment. LÖVE provides another layer to make things simpler in the shape of callback functions that will trigger when a collision occurs.

There are four callback functions used for collisions:

1. `beginContact` gets called when two objects collide.
2. `endContact` gets called when two objects stop colliding, say, when one bounces off the other and both objects stop touching each other.
3. `preSolve` is called just after a collision is detected but before the programmed action that executes automatically after the collision runs. In general, the default action is that, when one body hits another, they bounce off each other. You don't have to program this explicitly; it's just what LÖVE's physics engine does. But in the body of the `preSolve` function, you can change that to make, for example, the bodies smoosh together under certain circumstances or break into pieces Asteroids-style.
4. `postSolve` is called after the collision and is usually used to collect data of the collision's effect, like what direction each object is now headed and at what speed.

Setting up the callbacks is a simple task, just add

```
world:setCallbacks (beginContact, ➥
  endContact, preSolve, postSolve)
```

to your `love.load ()` function. While you are at it also add

```
tank_hit = false
```

to the function. You will use the `tank_hit` variable later to record when the tank gets hit by the mine if they both collide.

Now add what you see in Listing 4 to the end of `main.lua`.

As you can see, you are only going to worry about when two objects collide – to be precise, whether the tank and mine collide. The thing is, `function beginContact ()` triggers when *any* two objects collide. The tank is colliding with the ground all the time. The mine collides with the ground when it bounces. This could get confusing if you act on every time any object collides with any other object.

The `a` and `b` parameters in the function's parameter list contain the fixtures of the bodies that are colliding. Remember that a body's fixture [8] contains things like its shape, mass, degree of bounciness, and so on. You can also define your own attribute. For that you use the `Fixture:setUserData ()` function. On lines 9 and 22 in Listing 3, you are giving each fixture a short name, `"Mine"` and `"Tank"`, which you can then use on line 2 of Listing 4. The `if` statement will determine whether the objects colliding are `"Mine"` and `"Tank"` and, if they are, will set `tank_hit` to `true`.

You can then use that information in your `update` function to do something (line 5, Listing 5). In this case, if the mine hits the tank, the program exits immediately (Figure 5).

Listing 4: Collision Callbacks (Part of main.lua)

```
01 function beginContact (a, b, coll)
02   if (a:getUserData () == "Tank" or b:getUserData () == "Tank") and
        (a:getUserData () == "Mine" or b:getUserData () == "Mine") then
03     tank_hit = true
04   end
05 end
06
07 function endContact (a, b, coll)
08 end
09
10 function preSolve (a, b, coll)
11 end
12
13 function postSolve (a, b, coll, normalimpulse, tangentimpulse)
14 end
```

Listing 5: love.update (Part of main.lua)

```
01 function love.update (dt)
02   world:update(dt)
03
04   if tank_hit then
05     love.event.push('quit')
06   end
07 end
```

User Input

It may be fun to watch the mine land on the tank, but games are meant to be interactive! At some point, you are going to have to grab the input from the player and use it to affect the outcome of the game. Let's do that now, and, while we're at it, let's turn our sample code into a proper game (sort of) with a goal.

Using the elements you already have (a mine, a tank, some physics, and collision detection), let's make a game where the player must launch the mine from the left of the screen, over the mountain, in hopes of hitting the tank.

Start by defining four new variables in your `love.load ()` function: `aiming = true`, `fire = false`, `angle = 0`, and `force = 0`. You will use the `aiming` variable in your `love.update` function to read in key presses that aim your mine. The `fire` variable tells your program when the fire button (the space bar – see below) is pressed, which is the moment to launch the mine. The `angle` variable is the angle at which you will launch your mine, and `force` is the variable that sets the strength at which you will launch it.

Now take a look at Listing 6, which shows the `love.update` function.

The `love.keyboard.isDown ()` function checks that the key you pass as a parameter is pressed. You use the left and right arrow keys to move the mine left and right, the up and down keys to change the angle (from 0 degrees, straight ahead, to 90 degrees, straight up). The + and - keys increase or decrease the force from between 0 and 1,000, and you use the space bar to fire.

Once the player presses the space bar, the aiming phase ends (you set the `aiming` variable to `false`), and the fire phase starts (you set the `fire` variable to `true`).

The fire phase (lines 23 to 26) is very simple: You calculate the horizontal and vertical components of the force using basic trigonometry and apply it to the mine. You may reasonably assume that the LÖVE function you need to move the mine is `body:applyForce ()` [9], but this is more appropriate when you want to apply a force over several game cycles, like when you are accelerating a car or firing the boosters on a rocket. The thing you need here is `body:applyLinearImpulse ()` [10], which applies a force for an instant and then lets go.

As with `body:applyForce ()`, `body:applyLinearImpulse ()` takes the horizontal and vertical component of a force to accelerate the body in a certain direction. You can also add where on the body you want to apply the force, thus giving it a spin, but you don't need it here.

Making a few modifications to your `draw` function (Listing 7), you can show your player what angle they will fire at and the force.

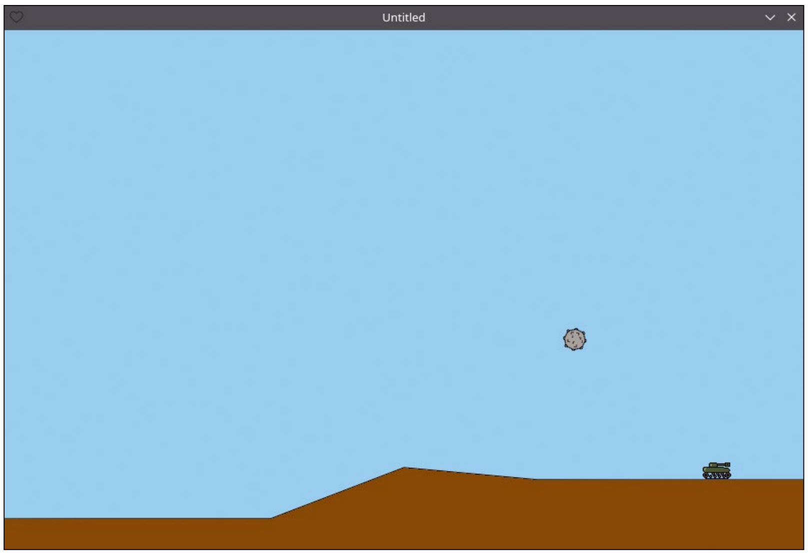


Figure 5: Will the mine hit the tank?

Listing 6: update (Part of main.lua)

```
function love.update (dt)
  world:update(dt)

  if aiming then
    if love.keyboard.isDown("right") and mine.body:getX () < 286 then
      mine.body:setX (mine.body:getX () + 1)
    elseif love.keyboard.isDown("left") and mine.body:getX () > 15 then
      mine.body:setX (mine.body:getX () - 1)
    elseif love.keyboard.isDown("up") and angle < 90 then
      angle = angle + 1
    elseif love.keyboard.isDown("down") and angle > 0 then
      angle = angle - 1
    elseif love.keyboard.isDown ("+") and force < 1000 then
      force = force + 1
    elseif love.keyboard.isDown("-") and force > 0 then
      force = force - 1
    elseif love.keyboard.isDown("space") then
      aiming = false
      fire = true
    end
  end

  if fire then
    mine.body:applyLinearImpulse (math.cos (math.rad (angle)) *
                                  force, math.sin (math.rad (angle))
                                  * force)
    fire = false
  end

  if tank_hit then
    love.event.push('quit')
  end
end
```

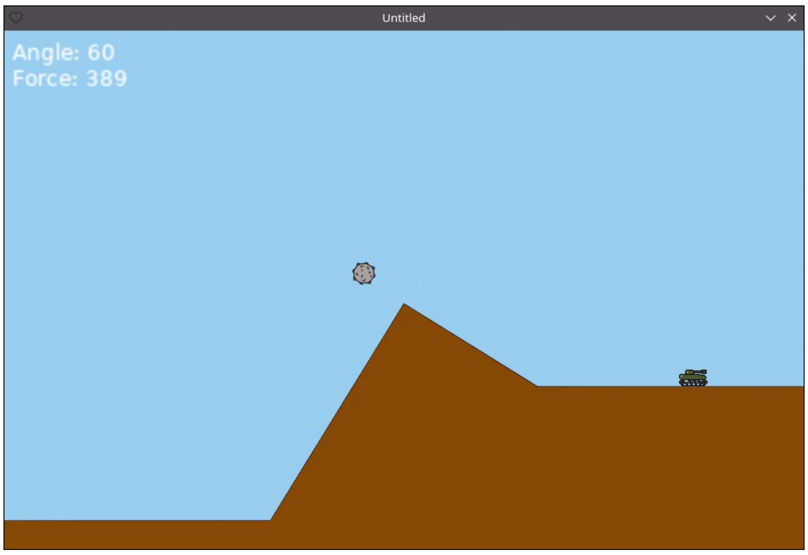

Untitled
Angle: 60
Force: 389

Figure 6: A proper game: Try to hit the tank!

Listing 7: draw (Part of main.lua)

```
function love.draw ()
  love.graphics.setColor(1, 1, 1, 1);
  love.graphics.print ('Angle: ' .. angle ,
                      10, 10, 0, 2)
  love.graphics.print ('Force: ' .. force ,
                      10, 40, 0, 2)

  mine:draw ()
  tank:draw ()

  terrainG:draw ()
end
```

As a side note, you have to know that there are many more ways of getting a player's input. LÖVE supports key presses, mouse movements and clicks, joysticks, gamepads, and touch screens [11].

The final game looks like what you can see in Figure 6.

Conclusion

Animations, game physics, collision detection, and player input handling should give you all the tools you need to create your own game. All you need now is an idea and a bunch of talent to make it a reality.

Have fun! ■■■

Info

[1] LÖVE 2D: *https://love2d.org*

[2] Lua: *https://www.lua.org/*

[3] "A LÖVE animation primer," by Paul Brown, *Linux Magazine*, issue 234, May 2020, pg. 88: *https://www.linux-magazine.com/Issues/2020/234/l-3-Animation*

[4] "Implementing physics in a LÖVE game," by Paul Brown, *Linux Magazine*, issue 235, June 2020, pg. 90: *https://www.linux-magazine.com/Issues/2020/235/Gravity*

[5] draw: *https://love2d.org/wiki/love.graphics.draw*

[6] Repository for the code in this article: *https://gitlab.com/linux-magazine/237/-/tree/master/*

[7] Contact objects: *https://love2d.org/wiki/Contact*

[8] Body fixtures: *https://love2d.org/wiki/Fixture*

[9] body:applyForce: *https://love2d.org/wiki/Body:applyForce*

[10] body:applyLinearImpulse: *https://love2d.org/wiki/Body:applyLinearImpulse*

[11] Input callbacks: *https://love2d-community.github.io/love-api/*

LINUX NEWSSTAND

Order online:
https://bit.ly/Linux-Newsstand

Linux Magazine is your guide to the world of Linux. Monthly issues are packed with advanced technical articles and tutorials you won't find anywhere else. Explore our full catalog of back issues for specific topics or to complete your collection.

#236/July 2020

Smarter Directories

The rigid structure of nested files and directories used on computer systems around the world was created more than 60 years ago, and experts believe we can do better. This month, you'll learn about some scripts for semantic file tagging in Linux.

On the DVD: Fedora Workstation 32 and Ubuntu "Focal Fossa" Desktop 20.04 LTS

#235/June 2020

What's New in Systemd

Systemd is a mystery that keeps on giving. Now a new feature of the leading Linux init system will change the way you think about user home directories. This month we take a closer look at systemd-homed.

On the DVD: Knoppix 8.6.1 and OpenMandriva Lx Plasma 4.1

#234/May 2020

Edge Computing

The Edge is a popular buzz word in high-tech news, but what does it mean really? We introduce you to an exciting new technology that could be changing the way we think about the cloud.

On the DVD: Manjaro 19.02 Gnome Edition and SystemRescueCd 6.1

#233/April 2020

Stream to Your TV

The line between computers and television blurred long ago, but the new tools and new ideas keep coming. This month we highlight some innovative apps for multimedia in Linux, including Gnome Cast for TV, and the easy-to-use Serviio media server.

On the DVD: The Complete Raspberry Pi Geek Archive

#232/March 2020

Stop Ads

Browser-based ad-blockers are useful for controlling many types of pop-ups and banners, but they are less effective with ads built into applications. We look at a couple of alternative tools for blocking ads at the network level: Pi-hole and Privoxy.

On the DVD: GParted 1.0.0 and Kali Linux 2019.4

#231/February 2020

Tiling

Are tiling window managers still relevant for today's desktop? We explore the possibilities of the tiling paradigm with a modern Linux built for tiling.

On the DVD: MX Linux MX-19 and Zorin OS 15 Core

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 *https://www.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*.

NOTICE

Be sure to check the event website before booking any travel, as many events are being canceled or converted to virtual events due to the effects of COVID-19.

KubeCon + CloudNativeCon 2020

Date: August 17–20, 2020

Location: Virtual Event

Website: *https://bit.ly/kube-cloudnativecon*

The Cloud Native Computing Foundation's flagship conference gathers adopters and technologists from leading open source and cloud native communities virtually from August 17-20, 2020. Join the community for four days to further the education and advancement of cloud native computing.

DevOpsCon

Date: August 31–September 3, 2020

Location: London, United Kingdom

Website: *https://devopscon.io/london/*

Join DevOpsCon to learn about the latest tools, technologies, and methodologies for building and maintaining secure, scalable, and resilient software systems. At DevOpsCon, you will meet internationally recognized thought leaders of the DevOps movement and benefit from their expertise.

Events

DrupalCon Global 2020	July 14-17	Everywhere	https://events.drupal.org/global2020/
KubeCon + CloudNativeCon Virtual	August 17-20	Virtual Event	https://bit.ly/kubecon-cloudnativecon
DevOpsCon	August 31-Sept 3	London, United Kingdom	https://devopscon.io/london/
Open Source Summit Japan	September 15-16	Tokyo, Japan	https://bit.ly/open-source-japan
Storage Developer Conference	September 21-24	Santa Clara, California	https://www.snia.org/events/storage-developer
Drupal GovCon 2020	September 28	Virtual Event	https://www.drupalgovcon.org/
Open Networking & Edge Summit North America	September 28-29	Virtual Event	https://bit.ly/Edge-summit
Smart Grid Cybersecurity 2020	October 6-8	Berlin, Germany	https://bit.ly/smart-grid-cybersecurity
openSUSE + LibreOffice Conference	October 15-17	Everywhere	https://events.opensuse.org/conferences/oSLO
All Things Open	October 18-20	Virtual Event	https://2020.allthingsopen.org/
Open Source Summit Europe	October 26-28	Dublin, Ireland	https://bit.ly/open-source-europe
DevOpsDays Berlin	October 28-29	Berlin, Germany	https://devopsdays.org/events/2020-berlin/welcome/
KVM Forum	October 28-30	Dublin, Ireland	https://bit.ly/KVM-forum
Cloud Foundry Summit Europe	October 29	Dublin, Ireland	https://bit.ly/cloud-foundry-summit
Linux Security Summit	October 29-30	Dublin, Ireland	https://bit.ly/Linux-Security-Europe
TechWeek Franfurt 2020	November 4-5	Frankfurt, Germany	https://www.techweekfrankfurt.de/

Images © Alex White, 123RF.com

CALL FOR PAPERS

We are always looking for good articles on Linux and the tools of the Linux environment. Although we will consider any topic, the following themes are of special interest:

- System administration
- Useful tips and tools
- Security, both news and techniques
- Product reviews, especially from real-world experience
- Community news and projects

If you have an idea, send a proposal with an outline, an estimate of the length, a description of your background, and contact information to *edit@linux-magazine.com*.

The technical level of the article should be consistent with what you normally read in *Linux Magazine*. Remember that *Linux Magazine* is read in many countries, and your article may be translated into one of our sister publications. Therefore, it is best to avoid using slang and idioms that might not be understood by all readers.

Be careful when referring to dates or events in the future. Many weeks could pass between your manuscript submission and the final copy reaching the reader's hands. When submitting proposals or manuscripts, please use a subject line in your email message that helps us identify your message as an article proposal. Screenshots and other supporting materials are always welcome.

Additional information is available at: *http://www.linux-magazine.com/contact/write_for_us.*

NOW PRINTED ON recycled paper from 100% post-consumer waste; no chlorine bleach is used in the production process.

Authors

Bernhard Bablok	58, 66
Erik Bärwaldt	16, 24, 52, 74, 80
Paul Brown	90
Zack Brown	12
Bruce Byfield	6, 28, 54
Joe Casad	3
Mark Crutch	71
Jon "maddog" Hall	72
Charly Kühnast	38
Christoph Langner	46
Vincent Mealing	71
Graham Morrison	84
Gaurav Nawani	76
Dmitri Popov	34
Mike Schilli	30
Mayank Sharma	40
Ferdinand Thommes	20, 46
Jack Wallen	8

Contact Info

Editor in Chief
Joe Casad, jcasad@linux-magazine.com

Copy Editors
Amy Pettle, Megan Phelps

News Editor
Jack Wallen

Editor Emerita Nomadica
Rita L Sooby

Managing Editor
Lori White

Localization & Translation
Ian Travis

Layout
Dena Friesen, Lori White

Cover Design
Lori White

Cover Image
© Iurii Kovalenko, 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 © 2020 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, Zieblandstr. 1, 80799 Munich, Germany.

Approximate	
UK / Europe	Aug 08
USA / Canada	Sep 04
Australia	Oct 05
On Sale Date	

Issue 238 / September 2020

Performance Tuning

Even the experts don't have a magic formula for performance tuning. Hundreds of parameters and settings have an effect on your system's performance. Next month we explore some options for dialing up the speed and efficiency of your Linux.

Image © chachar, 123RF.com

Preview Newsletter

The Linux Magazine Preview is a monthly email newsletter that gives you a sneak peek at the next issue, including links to articles posted online.

Sign up at: ***https://bit.ly/Linux-Update***

Shop the Shop shop.linuxnewmedia.com

DISCOVER LibreOffice

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

Explore the FREE office suite used by busy professionals around the world!

Create your own:

- **Word processing docs**
- **Spreadsheets**
- **Presentations**
- **Databases**

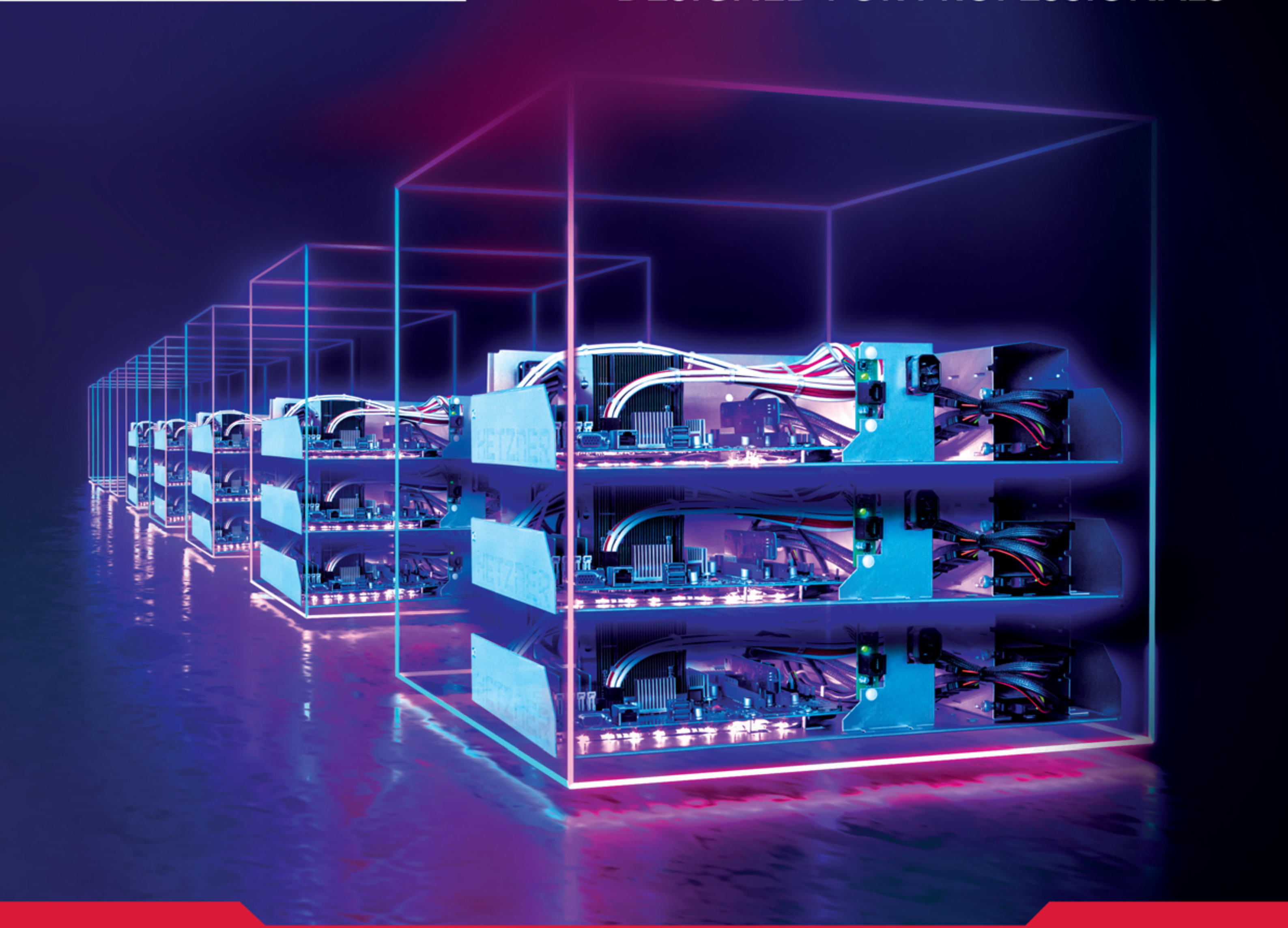
Whether you work on a Windows PC, a Mac, or a Linux system, you have all you need to get started with LibreOffice today. This single-volume special edition will serve as your guide!

Order online:
shop.linuxnewmedia.com/specials

For Windows, macOS, and Linux users!

HETZNER

DEDICATED ROOT SERVER
DESIGNED FOR PROFESSIONALS



Dedicated Root Server AX61-NVMe

- AMD Ryzen 9 3900 12-Core “Matisse” (Zen2) Simultaneous Multithreading
- 128 GB DDR4 ECC RAM
- 2 x 1.92 TB NVMe SSD
- 100 GB Backup Space
- Traffic unlimited
- Location Finland
- No minimum contract
- Setup Fee $96.50

monthly $91.50

Dedicated Root Server AX161

- AMD EPYC 7502P 32-Core “Rome” (Zen2) Simultaneous Multithreading
- 128 GB DDR4 ECC RAM (configurable, at additional cost)
- Individual storage capacity (at additional cost)
- 100 GB Backup Space
- Traffic unlimited
- Location Germany
- No minimum contract
- Setup Fee $129.00

monthly from $129.00

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

www.hetzner.com