Automated Login for Netflix and other sites

FREE DVD

LINUX PRO MAGAZINE

ISSUE 249 – AUGUST 2021

Turn Your Android into a Linux PC

without voiding the warranty

Maker Tricks

Build an electronic scale from scratch

PLUS:

- Loki: Easy tool for managing log data
- Track your documents with SeedDMS

5 Alternative Search Engines

LINUX NEW MEDIA
The Pulse of Open Source

WWW.LINUXPROMAGAZINE.COM

A Linux Distribution For Professionals

openSUSE Leap 15.3

Leap is a desirable distribution for IT professionals, entrepreneurs, hobbyists, small businesses and educational practitioners

get.opensuse.org

BRAIN RIGHTS

Dear Reader,

The NeuroRights Initiative has been in the news recently, and it seems like a good time for a shout out and a sincere thanks for their work. The initiative, which is hosted at Columbia University and headed by Columbia Neuroscience Professor Rafael Yuste, is a collection of scientists with the mission to raise awareness about human rights issues that could arise from the evolution of technologies targeting the human brain.

According to the NeuroRights Initiative website, "Neurotechnology, especially when paired with artificial intelligence, has the potential to foundationally alter society. In the coming years, it will be possible to decode thought from neural activity or enhance cognitive ability by linking the brain directly to digital networks. Such innovations could challenge the very notion of what it means to be human." [1]

Scientists have already used brain-computer interfaces (BCIs) to transcribe human thoughts, plant artificial memories in mouse brains, and control the actions of laboratory animals. If used humanely, these technologies could have enormous potential to assist people with disabilities or help patients overcome post-traumatic stress. However, if these technologies go unregulated and become tools for government or commercial control, the results could be devastating.

One of the goals of the NeuroRights initiative is to add five basic neuro-rights to the Universal Declaration of Human Rights, which was ratified by the UN in 1948. The five rights are:

1 . The right to identity, or the ability to control both one's physical and mental integrity;
2. The right to agency, or the freedom of thought and free will to choose one's own actions;
3. The right to mental privacy, or the ability to keep thoughts protected against disclosure;
4. The right to fair access to mental augmentation, or the ability to ensure that the benefits of improvements to sensory and mental capacity through neurotechnology are distributed justly in the population;
5. The right to protection from algorithmic bias, or the ability to ensure that technologies do not insert prejudices.

These five tenets seem quite reasonable and obvious to us in our early-21st-century comfort zone, but you can imagine how each of these rights could easily come under fire as the forces of money and power descend on the neuro space. It is particularly important to draw some boundaries now, before venture capitalists settle in, because the chance for meaningful legislation diminishes when big money, market forces, and paid lobbyists line up to oppose it.

This tech is still at a state where only the scientists understand it, so it helps when this kind of early warning comes from the scientists. Whether the warning is soon enough, only time will tell, but the clear and thoughtful voice of the NeuroRights Initiative gives me hope that scientists in other fields, such as biotech, nanotech, and AI will take a similar interest in laying the groundwork for humane use of their technology.

The NeuroRights Initiative and its partners are also interested in building a culture of ethical accountability for technologists by developing a *Technocratic Oath*, modeled on the Hippocratic oath taken by doctors, that will serve as "…an ethical framework for entrepreneurs, physicians, and researchers developing Neurotechnology and AI." [2]

Ultimately, though, the only way to protect ourselves from the dangers of neuro tech is through real laws passed at the national level by real governments with powers of enforcement and oversight. The need for clarity on this important issue is yet another reason why we need governments that actually communicate and cooperate to pass meaningful legislation, rather than just tap dancing at fundraisers and gesturing to their base.

It seems that now, more than ever, the future depends on the questions we are answering today.

Joe

Joe Casad,
Editor in Chief

Info

[1] NeuroRights Initiative: *https://nri.ntc.columbia.edu/*

[2] Technocratic Oath (Coming Soon): *https://nri.ntc.columbia.edu/content/technocratic-oath-coming-soon*

LINUX MAGAZINE
AUGUST 2021

ON THE COVER

NEWS

COVER STORY

REVIEWS

IN-DEPTH

MakerSpace

LINUXVOICE

TWO TERRIFIC DISTROS
DOUBLE-SIDED DVD!
SEE PAGE 6 FOR DETAILS

openSUSE Leap 15.3 and Kubuntu 21.04

Two Terrific Distros on a Double-Sided DVD!

openSUSE Leap 15.3

64-bit

OpenSUSE Leap 15.3 includes increased build consistency and seamless migration to the commercial SUSE Linux Enterprise Server and the latest array of openSUSE's developing set of artificial intelligence packages. Those working with containers may also want to check out Podman, a daemonless container engine for developing, managing, and running OCI Containers, and kubeadm, a tool for creating minimal, best-practices Kubernetes clusters.

For everyday users, Leap 15.3 features the usual desktop environments. For KDE users, it includes the long-term support Plasma 5.18, whose new features include a built-in emoji picker, strong GTK support for those who like to mix their desktops and applications, and better support for public WiFi. Also supported is Gnome 3.34, which includes visual makeovers for icons, login, and the overview screen; sandboxed browsing; and drag-and-drop actions for improving folder management. The installer also includes an option for Xfce 4.16, which features extensive cosmetic innovations.

As always, openSUSE Leap is a distro for all levels of users, with something for everyone.

Kubuntu 21.04

64-bit

Kubuntu is the Ubuntu flavor that installs with KDE Plasma as the default desktop environment. The 21.04 releases of all Ubuntu flavors are interim releases, which means that they contain more refinements than major changes, and, in most ways, Kubuntu is not an exception. However, while other Ubuntu flavors default to Wayland, Kubuntu does not, although users can switch to Wayland at login. Be aware, though, that Wayland is not officially supported by Kubuntu and is included for experimental purposes only. The main reason for this difference is that Wayland support for Plasma lags behind the support for Gnome and other major desktops.

What Kubuntu 21.04 does feature is Plasma 5.21, the latest stable version of the desktop. Many of the improvements in Plasma 5.21 are behind the scenes, but users might want to explore the new system monitor, as well as the other updates to standard apps, such as LibreOffice 7.1 and Firefox 87. Kubuntu makes a habit of including the latest applications, especially for KDE, and is an ideal place to start learning about KDE Plasma.

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.

DrupalCon
EUROPE2021
Sponsor

For Drupal's 20th Anniversary, more than ever, show your support to the Drupal project, become an official sponsor!

THE SPONSORSHIP BROCHURE IS OUT, FULL OF NEW OPPORTUNITIES TO BE DISCOVERED!

Secure your package now, visit
https://events.drupal.org/europe2021/become-sponsor

For more information, drop us an email at
drupal@kuoni-congress.com

NEWS

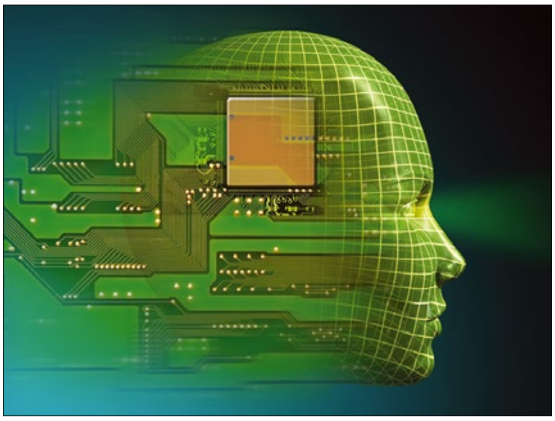
Updates on technologies, trends, and tools

THIS MONTH'S NEWS

NVidia and Valve Collaborate to Bring DLSS to Linux

DLSS stands for Deep Learning Super Sampling and is a temporal image upscaling technology developed by NVidia for NVidia graphics cards. This technology makes it possible to upscale lower-resolution images into higher-resolution images so they can be better displayed on higher-resolution displays.

This technology would greatly enhance the Linux gaming experience but isn't available yet for the open source platform. Both NVidia and Valve are working to change that. By using Tensor Cores on RTX GPUs to accelerate performance for supported games, NVidia is making an effort to bring DLSS to Steam Proton on Linux. This means games such as Call of Duty, Cyberpunk 2077, Fortnite, and Rainbow Six Siege could soon be playable on Linux, via Steam.



© Andrea Danti, 123rf.com

This technology will require supported graphics cards, such as the GeForce RTX 3080 and the upcoming NVidia 470 (which will also have Wayland support).

In a press release, NVidia announced (during their keynote for Computex 2021, *https://www.youtube.com/watch?v=1Kio8Hn8f3U*) that it is "collaborating [with Valve and the Linux gaming community] to bring NVidia DLSS to Proton – Linux gamers will be able to use the dedicated AI cores on GeForce RTX GPUs to boost frame rates for their favorite Windows Games running on the Linux operating system."

Kali Linux 2021.2 Official Release Now Available

If you're a fan of Linux and security, you'll be glad to know the developers of Kali Linux have released a new version of the fan-favorite distribution. Kali Linux 2021.2, created by Offensive Security, includes a new collection of themes and plenty of features to attract cybersecurity pros, ethical hackers, and anyone else interested in a user-friendly penetration testing platform.

One of the most anticipated additions to Kali Linux is Kabox, which is a tool for packaging tricky applications in Docker containers to be used in the distribution. These apps are normally a challenge to package because of dependencies, legacy libraries and programs, and applications that require isolation to run.

Another new addition to Kali Linux is Kali-Tweaks, which is an automation tool aimed at helping users easily customize the operating system for tasks such as installing or removing groups of tools, changing the default login shell, and enabling different repositories (such as those for bleeding-edge and experimental software).

Also included with the latest version of Kali Linux are new packages for Raspberry Pi, new Kali-Docker images, support for Apple M1 hardware (on Parallels), Kali NetHunter support for Android 11, Ghidra (a reverse engineering tool created by the National Security Agency), and plenty of updates to the look and feel of the desktop.

Get a copy of Kali Linux 2021.2 for your platform of choice from the official download page *(https://www.kali.org/get-kali/)*.

Entroware Unleashes a Beast of a Linux Laptop

Are you on the lookout for a Linux-powered laptop that can go the distance and exceed your needs for power, size, and battery life? If that sounds like the perfect machine to fill your dance card, and you're not looking for the most brilliant display on the market, Entroware has the laptop for you.

The Proteus 15.6-inch is a redesign of the original mobile workstation and includes something few other Linux laptops can match – a 73 Wh battery that can be charged via USB Type-C cable (so you don't have to carry around so many cables). The original Proteus included a 62.16 Wh battery, so this is a good jump in battery life. Although there is no claim about battery life to be found on the Proteus page (*https://www.entroware.com/store/proteus*), you can expect to be able to use this laptop for quite some time before you're plugging it back in for a charge.

© Alphaspirit, 123rf.com

The battery isn't the only impressive feature: The Proteus base model ships with a Core i5-1135G7 (up to 4.20 GHz with 4 Cores and 8 Threads), 8GB of DDR4 2666 MHz RAM, and a 250GB PCIe NVMe SSD. The Proteus can be maxed out with a Core i7, 64 GB of RAM, and up to a 2TB SSD. And although the screen isn't all that brag-worthy, it's still an IPS LED at 1920x1080). The Proteus also includes an expansion slot for M.2 storage devices, which allows you to add up to 4TB of capacity.

The base price of the Proteus is £819.99. You can purchase the Proteus now through the Entroware website (*https://www.entroware.com/store/proteus*).

System76 Unveils Its "Launch" Keyboard

System76 loves to push every envelope possible. They've created one of the finest desktop computers on the market, have an outstanding lineup of laptops, and their servers are all top-notch. Soon, users will also be able to purchase an open source keyboard called Launch.

The Launch keyboard is a highly customizable piece of hardware with firmware based on QMK. The keyboard even includes a space bar that's been split into two keys so that you can customize one of them to serve a different purpose. Speaking of keys, the Launch is lit with RGB lights that can also be customized.

The Launch is milled out of a single piece of aluminum and features a high-speed USB hub, easy key swapping, personalized application shortcuts, tenkeyless (TKL) layout (for a smaller footprint), a choice between Jade and Royal switches (for either a muted or obvious click), and an included magnetic foot to give the keyboard a 15 percent incline (for those who need more ergonomics).

MORE ONLINE

Linux Magazine

www.linux-magazine.com

ADMIN HPC

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

When I/O Workloads Don't Perform
- Petros Koutoupis

Every now and then, you find yourself in a situation where you expect better performance from your data storage drives. Either they once performed very well and one day just stopped, or they came straight out of the box underperforming. I explore a few of the reasons why this might happen.

Preload Trick
- Jeff Layton

As they gain experience, administrators learn a topic system called the "LD_PRELOAD Trick," which can help fix misbehaving applications, upgrade applications, and even improve application performance. Of course, it's not really a trick, just the use of a feature in *nix operating systems.

ADMIN Online

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

Cyber Threat
- Matthias Wübbeling

The danger of ransomware attacks calls for a robust backup and monitoring strategy.

Access Granted
- Matthias Wübbeling

Log in to your account securely without a password with LDAP and a schema to establish the objects and attributes required for FIDO2 authentication.

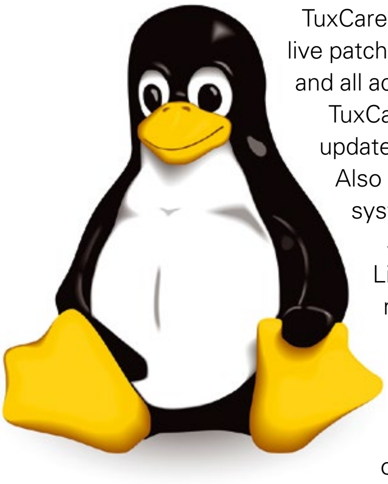
Perfect Harmony
- Thorsten Scherf

With the integration of Podman and systemd, you can put any software inside a container under the control of systemd and see almost no difference between running the service directly on the host or inside a container.

The Launch is 100 percent open source and is compatible with Linux, macOS, and Windows. Linux users will be able to update the firmware from within Pop!_OS. You can pre-order your Launch now (*https://system76.com/accessories/launch_1/configure*) from System76, starting at $285.00.

CloudLinux Launches TuxCare

CloudLinux recently unleashed AlmaLinux and has made good on its promise to offer enterprise-level support for the CentOS replacement. This new service, called TuxCare (*https://tuxcare.com/*), is an expansion of the CloudLinux KernelCare and Extended Lifecycle Support service brands and makes it easy for businesses to purchase support packages for their Linux deployments. TuxCare will cover support, maintenance, and security for enterprise Linux systems.



TuxCare rolls in what CloudLinux used to call KernelCare, which offers live patching for critical components (from the kernel to shared libraries and all across the Linux stack).

TuxCare's Linux Support Services will include regular patches and updates that will be delivered for every enterprise Linux component. Also included will be 24/7 incident support (even with operating systems that have reached EOL).

Jim Jackson, president and chief revenue officer of CloudLinux, said of TuxCare, "Building out a larger brand to house our rapidly expanding set of services makes it easier for our customers and prospects to see everything we provide in one place." Jackson continued, "Under the new TuxCare umbrella customers can review and select everything they need from our cohesive collection of services to take care of their Linux infrastructure."

In the future, CloudLinux plans on including databases and virtualization stack live patching into the service.

For TuxCare pricing, check out the official price/feature matrix (*https://tuxcare.com/pricing/*).

AlmaLinux Releases 8.4

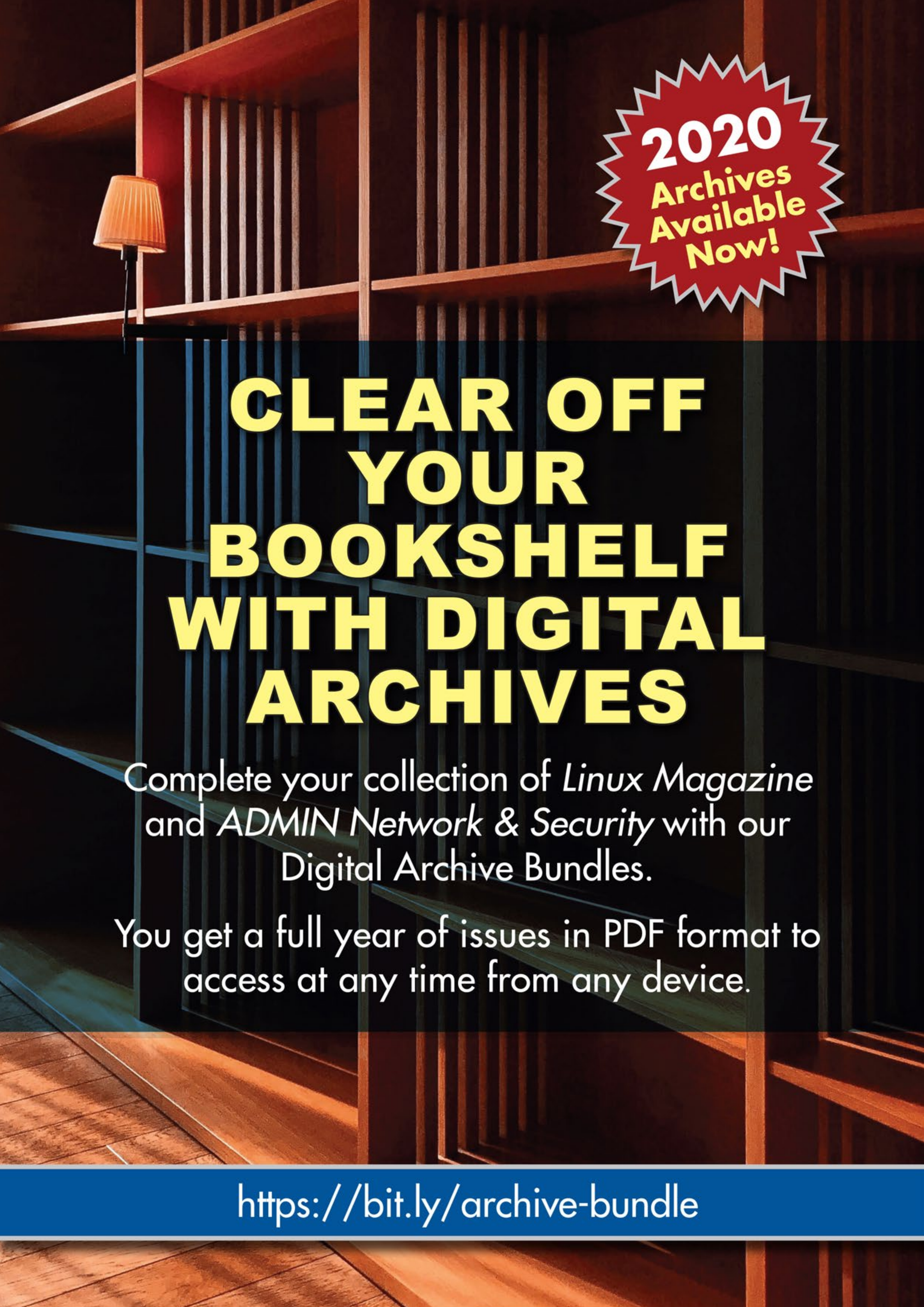
To stay in step with Red Hat Enterprise Linux (RHEL) 8.4, the AlmaLinux OS Foundation has officially released AlmaLinux 8.4. This enterprise-grade 1:1 binary replacement for RHEL is the second stable release of AlmaLinux. The biggest feature added to the latest release is support for Secure Boot.

Other features found in the latest release include support for OpenSCAP, the release of a devel repo (which includes extra packages and build dependencies that are not included in the upstream distribution), a few new module streams (such as Python 3.9, SWIG-4.0, Subversion 1.14, Redis 6, PostgreSQL 13, and MariaDB 10.5), and updates to compilers (such as GCC Toolset 10, LLVM Toolset 11.0.0, Rust Toolset 1.49.0, and Go Toolset 1.15.7).

Another change is that the PowerTools repo has been disabled by default (to match upstream distribution) and moved to a separate configuration file (almalinux-powertools.repo).

Jack Aboutbouls, community manager of AlmaLinux, said about AlmaLinux 8.4., "Our community, partners and sponsors all worked together, at warp speed no less, to provide the very latest, best and stable community-driven, governed and free open source alternative to the former incarnation of CentOS." Aboutbouls continued, "The open source world was nervously watching the past few months wondering how feasible this new reality would be, and for us to produce something suitable for real-world, actively running workloads so quickly and efficiently is truly a testament to our provenance, the power of community, and the open source model."

Download an ISO copy of AlmaLinux 8.4 from one of the many available mirrors (*https://mirrors.almalinux.org/*).



Zack's Kernel News

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

By Zack Brown

Protecting Filesystems from Themselves

Chao Yu recently tried to revert a kernel commit for an F2FS patch. F2FS is a Samsung filesystem for solid state drives. Chao wanted to revert the patch because one of the kernel's generic tests expected F2FS to fail to mount a read-only partition. Ironically, as pointed out by Jaegeuk Kim, F2FS had no trouble mounting such a partition and giving the user full read access to all its data. So the filesystem failed the test … because it succeeded.

Jaegeuk suggested changing the test rather than reverting the patch, but Chao pointed out that the test was actually important for filesystems in general, not just F2FS. Changing the test for that one case, he said, would mean other filesystems might technically pass the test when they really should fail.

Chao also disagreed with Jaegeuk that F2FS handled the case properly. Walking through the code, he identified a certain point at which, he said, the device was then read-only, so that all writes would fail. Therefore, recovered data would not be able to persist beyond the expiration of the page cache. At that point, he said, the user would see stale data instead of the latest system state.

Jaegeuk argued that – and this became problematic in the conversation – F2FS could synchronize its data in the device core, with the filesystem itself, if the user then mounted the filesystem as read-write. So there was no need for F2FS to fail the test, given that it could mount the filesystem and potentially also save new data.

At this point, Chao began to suspect they were talking about two different things.

There was the one case, where the filesystem is mounted with the read-only option. In that case, the user would be unable to give any command to write to that filesystem. The filesystem itself, on the other hand, would still retain the power to write to the underlying solid state device. So even if the user couldn't write anything, the filesystem would still be able to preserve some state, which was what Jaegeuk had been saying.

The second case, Chao pointed out, was where the device itself is read-only. In that case, there would truly be no way to preserve any state. This was the case Chao was concerned about. He agreed with Jaegeuk that the kernel test wasn't really relevant for the case where the filesystem was read-only while the device was read-write. The key case for Chao was when both the filesystem and the device were read-only. In that case, Chao said, the generic kernel test was correctly failing, which justified his effort to revert the particular patch in question.

Chao and Jaegeuk descended into a technical implementation discussion for future versions of Chao's patch, and eventually Chao came out with version 2.

The fun part of that debate is the effort to allow the user as much control as possible over their system. If there's some tiny remnant of data that might be preservable, the developers don't want to miss the chance to preserve it, even in odd scenarios where the user mounts a filesystem first as read-only and then as read-write.

Extending chroot() to Regular Users

Mickaël Salaün wanted to extend `chroot()` so that regular users also could use it. The `chroot()` system call creates a new root directory for the current process and its children. It's used in conjunction with lots of other things to create virtual systems that appear to be entirely distinct from the actual running system. Generally, only the root user does this, but Mickaël made the case that there was real value in letting regular users do it, too.

In particular, he wanted to let regular users create sandboxes (secure areas within a system) so they could develop

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**.

their projects without fear of hostile users taking advantage of momentary bugs.

Mickaël said, "`Chroot(2)` is not an access-control mechanism per se, but it can be used to limit the absolute view of the filesystem." And, he continued, "Users may not wish to expose namespace complexity to potentially malicious processes, or limit their use because of limited resources. The `chroot` feature is much more simple (and limited) than the mount namespace but can still be useful."

The security considerations in Mickaël's patch are real. As he put it, "Allowing a task to change its own root directory is not a threat to the system if we can prevent confused deputy attacks, which could be performed through execution of SUID-like binaries. This can be prevented if the calling task sets PR_SET_NO_NEW_PRIVS on itself with prctl(2). To only affect this task, its filesystem information must not be shared with other tasks, which can be achieved by not passing CLONE_FS to clone(2). A similar no_new_privs check is already used by seccomp to avoid the same kind of security issues. Furthermore, because of its security use – and to avoid giving a new way for attackers to get out of a chroot (e.g., using `/proc/<pid>/root`, or `chroot/chdir`) – an unprivileged chroot is only allowed if the calling process is not already chrooted. This limitation is the same as for creating user namespaces."

The goal, from Mickaël's perspective, would be to allow regular users to gain the benefits of using `chroot()` on their own, while protecting the larger system from any security issues that might then arise.

However, Mickaël's patch got a serious smackdown from Casey Schaufler. Casey said, "I don't see that new comments are necessary when I don't see that you've provided compelling counters to some of the old ones." Among other things, Casey felt that the namespaces feature would be sufficient for any of the use cases Mickaël had identified. The Linux namespace feature is a way to hide resources from processes. Like `chroot()`, namespaces create isolated areas that appear to the process to be an entire running system. Casey felt that namespaces would do the trick, and there was no need to extend `chroot()` to support regular users.

Mickaël disagreed. It's not that he felt namespaces couldn't accomplish the features. It was more that he felt it would be riskier. As he put it, "namespaces bring complexity which may not be required. When designing a secure system, we want to avoid giving access to such complexity to untrusted processes (i.e., more complexity leads to more bugs). An unprivileged chroot would be less complex. Of course it is not enough on its own, but it can be combined with existing (and future) security features."

Casey, for his part, felt that Mickaël's `chroot()` extensions would require users to behave in such a restricted way that the feature would be virtually unusable.

He also pointed out that if Mickaël's user restrictions were applied across the board, there were other solutions that would work just as well as Mickaël's idea that wouldn't require extending `chroot()`.

Casey just didn't buy the argument that namespaces were too complex. As he put it, "I can use a Swiss Army Knife to cut a string even though it has a corkscrew." And he said, "If you're *really* designing a secure system you can design it to use existing mechanisms, like CAP_SYS_CHROOT!"

Mickaël, however, countered, "Not always. For instance, in the case of a web browser, we don't want to give CAP_SYS_CHROOT to every user just because their browser could (legitimately) use it as a security sandbox mechanism. The same principle can be applied to a lot of use cases, e.g., network services, file parsers, etc."

Casey was unmoved. He said, "You've identified a clever hack to justify expanding when `chroot()` could be done 'safely' without using privilege. Why not learn how to use the existing mechanism properly? And teach the next set of people how to do the same? I am under no delusion that we can tweak here and fiddle there and make security all rainbows and unicorns. Mature mechanisms that are general are safer than tangled heaps of special cases that make individual projects easier."

However, in the midst of this seemingly mega-rejection, Casey also launched this odd little projectile, saying, "In any case, if you can get other people to endorse your change, I'm not all that opposed to it. I think it's gratuitous. It irks me that you're unwilling to use the facilities that are available and instead want to complicate the security mechanisms and policy further. But that hasn't seemed to stop anyone before."

At this point, the debate between the two ended abruptly. Instead, Kees Cook spoke up from the sidelines to say:

"The only part of this design that worries me is that it seems as though it's still possible to escape the `chroot` *if a process didn't set up its fds carefully, as Jann discussed earlier: https://lore.kernel.org/lkml/c7fbf088-02c2-6cac-f353-14bff23d6864@digikod.net/.*

"Regardless, I still endorse this change because it doesn't make things worse, since without this, a compromised process wouldn't need any tricks to escape a chroot because it wouldn't be in one. It'd be nice if there were some way to make future openat() calls be unable to resolve outside the `chroot`*, but I view that as an enhancement.*

"But, as it stands, I think this makes sense."

As far as Kees was concerned, the code could go directly into the kernel without delay. In terms of exactly who would accept the patch and feed it up to Linus Torvalds, Kees was not sure. He remarked, "If Al is too busy to take it, and James would rather not take VFS, perhaps akpm would carry it? That's where other similar VFS security work has landed."

Al Viro remarked, "Frankly, I'm less than fond of that thing, but right now I'm buried under all kinds of crap […]. I'll post a review, but for now it very definitely does *not* get an implicit ACK from me."

And that was the end of the discussion.

Rarely does a security-related patch get such a thrashing as Casey gave Mickaël's and still get immediately accepted into the kernel. It's still possible Al will raise a serious objection (in which case the patch would be a dead duck), or some other security concerns may come up. But if Kees is right, the main deciding factor could be that the patch doesn't make anything worse and could improve security in general. If a regular user made a mistake with `chroot()`, it would still only expose that regular user to attack. With no root user behind it all, there would be no serious reward at the end of that attack.

Tracking "Issues"

Thorsten Leemhuis recently proposed a new Linux kernel development mailing list, "linux-issues." The idea would be for developers to CC [carbon copy] their various problems to that list, which would then become a sort of central repository for all kernel-related issues.

The number of existing Linux kernel development mailing lists is truly uncountable. You can see a lot of them listed at *https://lore.kernel.org/lists.html*, but there are undoubtedly vast numbers of mailing lists used by small groups of kernel developers working in close collaboration. Many of those will also be behind corporate firewalls. Counting them all would truly be impossible.

The ones available at *lore.kernel.org* are archived and searchable. This was one of Thorsten's main ideas: a single searchable list for all kernel issues.

The idea itself was fairly flexible, based on discussions at various kernel conferences. Thorsten gave some background, saying, "Back on the maintainers summit in 2017 it was agreed to create a dedicated list for this purpose (*https://lwn.net/Articles/738216/*). I even requested a linux-regressions@vger.kernel.org a while later but didn't hear anything back; sadly, about the same time, I started having trouble finding spare time for working on regression tracking."

Thorsten tried to anticipate certain objections. For example, he said, "The question 'Why not simply LKML' [Linux kernel mailing list] will likely pop up, but the thing is, searching for reports there will often turn up patches that improve the kernel and don't fix anything. That makes it hard to find issue reports, especially for users that are not used to deal[ing] with mailing lists and their archives."

Thorsten added, "Yes, I'm quite aware that searching linux-issues@lists.linux.dev list obviously won't turn up reports that are filed in *bugzilla.kernel.org* or some other bug-tracking tool. That's okay, as the reporting-issues.rst tells users to look in those places as well."

And, Thorsten said, "reporting issues/bugs by mail has downsides, and maybe instead of creating yet another mailing list, it would be better if all the kernel issues would be reported to a central place like *bugzilla.kernel.org*. But that tracker doesn't work that well currently, as quite a few of the issues filed there, AFAICS [as far as I can see], never reach the people that need to be handle them. I don't see that changing any time soon (we had a discussion about this recently: *https://lore.kernel.org/linux-doc/20210111194822.4kvl2tx24anyu23k@chatter.i7.local/*)."

Lukas Bulwahn said he supported Thorsten's general idea, but Lukas wanted some clarity on what an "issue" really was. For example, would the list include all the automated build warnings, test bot warnings, and other automated kernel testing systems? Or, Lukas asked, "Would you like to keep this list only for reports from single individual human users that need to detect the 'issue' without using one of the tools above?"

Meanwhile, Konstantin Ryabitsev offered an update on the status of *lore.kernel.org*. He said:

"There will soon be a unified 'search all of lore.kernel.org regardless of the list/feed source' capability that may make it unnecessary to create a separate list for this purpose. There's active, ongoing work in the public-inbox project to provide parallel ways to follow aggregate topics, including query-based subscriptions (i.e., 'put a thread into my inbox whenever someone mentions my favourite file/function/device name'). This work is not complete yet, but I have great hopes that it will become available in the next little while.

"Once we have this ability, we should be able to plug in multiple sources beyond just mailing lists, including a feed of all bugzilla.kernel.org changes. This should allow someone an easy way to query specific bugs and may not require the creation of a separate list.

"I'm not opposed to the creation of a new list, of course – just want to make sure it's aligned with the improvements we are working to make available."

Thorsten replied, "Ahh, nice, thanks to everyone working on that!"

James Bottomley also replied to Konstantin, saying, "I suspect the problem is that there's no known useful search string to find a bug report even given a searchable set of lists, so the main purpose of this list would be 'if it's on here, it's a bug report', and the triage team can CC additional lists as appropriate. Then we simply tell everyone to send kernel bugs to this list and ask maintainers to CC it if a bug report shows up on their list?"

Linus Torvalds also had a suggestion for the new list. He said, "I'd much prefer the name 'linux-regressions' as being much more targeted than 'linux-issues'. Make it clear that the list is only for regressions that people can describe some way, rather than some general 'I have issues with xyz'. The more clear-cut the list is, the better, I think."

The discussion continued and ended inconclusively. Essentially, a broad range of developers offered their sense of the various corner cases and problems that might come up or that might be solved by various approaches to Thorsten's new list.

Some comments, such as those from Theodore Ts'o, were mostly hoping that regular users would read more documentation about how to submit bug reports. To that extent the discussion ranged beyond Thorsten's original question. Theodore, for example, said, "I wonder if it might be useful to have a form which users could be encouraged to fill out so that: (a) The information is available in a structured format so it's easier for developers to find the relevant information, (b) so it is easier for programs to parse, for easier reporting or indexing, and (c) as a nudge so that users remember to report critical bits of information such as the hardware configuration, the exact kernel version, which distribution userspace was in use, etc."

It's not at all clear what will come out of this discussion. Some new sort of bug tracker, certainly. There were many ideas floating around. It's also true that creating a new mailing list is an extremely noncritical operation. So it would not be weird to see one or two pop up, with rapidly changing goals and definitions, until the idea finally settles into something no one had expected at the start. ■■■

Running Linux apps on Android without rooting your phone

Phone as Linux PC

UserLAnd lets you run Linux applications on your Android phone – all without replacing Android OS.

By Dirk Ambras

Smartphones are getting more and more powerful. Is it time to start thinking of a smartphone as something more like a tablet or a real computer? Is it possible to set up an Android phone with a full version of Linux – connected to a monitor, keyboard, and mouse?

Android OS is based on a modified version of the Linux kernel. In theory, a hardware-compatible Linux should be able to run on systems that Android OS runs on, but the Android developers at Google have added lots of proprietary drivers and bits. Additionally, the complexity of the communication hardware means that Android has drifted quite a distance from what we think of as a conventional Linux. Still, the power of the GPL means that Android OS is available to the community in source code form [1], and of course, developers all over the world are always tinkering with Linux to make it do useful things. A number of independent distros have evolved to provide mobile alternatives based on both the Linux kernel and the Android kernel (see the box entitled "Linux on Android Distros").

Linux on Android Distros

Several Linux distributions bill themselves as replacements for Android in the mobile space. These distros are all at various stages of completion – some work for certain phones but not others, and some are still in preliminary stages of development. Be sure to check the hardware compatibility list and read as much as you can about these alternatives before you install. Examples include projects such as:

- Ubuntu Touch – a mobile version of Ubuntu. When Canonical backed off on their mobile initiative a few years ago, the UBports community took over the development and maintenance.
- postmarketOS – a mobile OS based on Alpine Linux.
- Sailfish OS – a commercial project that bills itself as "a European alternative to dominating mobile operating systems." Sailfish is based on the code from the old MeeGo mobile OS.
- Mobian – a project promising Debian for mobile devices.

Another class of distros retains the Android base system but removes the proprietary parts, thus creating what is basically an Android-based system that is free of Google. Distros such as LineageOS, Replicant, /e/, and GrapheneOS are examples of open source, non-proprietary systems based on Android.

A full install of a mobile-based Linux on your smartphone provides a complete escape from Google's walled garden and offers access to the wide range of applications provided with Linux. But it also comes with some risks. You'll need to root your phone to replace the factory-installed OS, which will almost certainly void the warranty. If anything goes wrong, you might not be able to restore the original system.

This article describes a less radical alternative for running Linux on your Android. UserLAnd is a compatibility layer for Android OS that lets you run selected Linux distros on Android – all without rooting you phone. Of course, you are not really replacing Android, so if that is your goal, one of the alternative Linux mobile systems would be a better choice. But if you're looking for the Linux desktop, UserLAnd delivers it with minimal disruption. I'll show you what worked and what didn't – and I'll explore how the performance of a UserLAnd Linux smartphone compares to an average home computer.

A note on the hardware used in this study: My smartphone is powered by a 2018 Qualcomm Snapdragon 845 with 8 cores and 4GB RAM, including an Adreno 630 GPU. I also tested my daughter's phone, with the successor Snapdragon 855 and 6GB RAM plus an Adreno 640 GPU.

Monitor

If you want to embark on the adventure of converting your smartphone into a PC, the first step is to equip it with the necessary components. First comes the monitor. An Android smartphone with a USB-C port can output the screen image on a monitor or TV. However, not all recent Androids support this option; some – by Sony, for example – fail to identify the adapter.

But that does not mean you have to give up. If you don't have a working port, you can always fall back on wireless screen sharing with Google Cast or another similar tool. I have also had some good experiences screensharing (also known as screen mirroring) with Miracast on LG and Sony devices. Miracast is a Wi-Fi Alliance standard that connects two devices wirelessly via Wi-Fi Direct and was integrated into Android versions from 4.2 and up to 6.0. Google's Chromecast took the place of Miracast in later Android versions.

Mouse and Keyboard

You can easily connect a mouse or keyboard via Bluetooth or with a suitable adapter. I use a combined USB-C Type-C-to-HDMI-HDTV adapter, which offers one HDMI, one USB-C, and one USB-A 3.0 port. Because the adapter gets very warm, I added a passive heat sink which was available from another project.

I plugged a small dual distributor into the USB-A port, which also has space for a microSD card. I had to make the distributor narrower on the side to make room for the wide HDMI connector. A small extension or a differently designed distributor would also solve the problem if necessary. There are also adapters that offer four independent USB-A ports.

Software

Android 10 comes with a new desktop mode that it enables when an HDMI adapter is connected. Depending on the manufacturer, this feature works more (Samsung, Huawei) or less (LG) well. You can also disable desktop mode in favor of simple screen mirroring. No matter what you decide on, the LG smartphones always output Full HD at 60Hz. Many Android apps still have to be adapted to this new desktop mode (wide format).

In order to run conventional Linux applications on Android, you need to emulate a Linux runtime environment on the smartphone. Luckily, the Linux community has a solution: the UserLAnd project, which is hosted on GitHub [2]. The software is also available on F-Droid and in the Google Play Store, or you could also compile it yourself – more on that later. Rooting is not required, and you do not need Google services.

The display of the smartphone's screen content is handled by a VNC viewer, which also runs on the smartphone and then sends the image to the external monitor (Figure 1). Various viewers are available for this; I can recommend bVNC Free and RealVNC. The VNC session runs in the smartphone on the localhost address 127.0.0.1:5951.

To completely fill the image area of the external monitor, you might have to set the format to 16:9. RealVNC v3.6.1 cannot cope with the Android 10 desktop mode at all, and bVNC always outputs an overly bright image; however, bVNC offers a well-scaled mouse pointer, at least in desktop mode. If the desktop is not completely visible or you want to enlarge a part of it, you can adjust with a two-finger gesture on the smartphone.

UserLAnd

The UserLAnd website offers the following description: "When UserLAnd first launches, it presents a list of common distributions and Linux applications. Clicking one of these then leads to a series of set-up prompts. Once these have been completed, UserLAnd will download and set up files required to start the task that has been selected. Based on the set-up, you will then be connected to your Linux distribution or application in a terminal or VNC viewing Android application."

UserLAnd offers a user-friendly (basic) installation of distributions such as Alpine, Arch, Debian, Kali, and Ubuntu. In addition, you can choose a desktop with LXDE or Xfce or just single applications like Firefox, LibreOffice, Octave, Gimp, and others.

The LXDE desktop environment is a good place to start, and it already comes with Firefox. If you were to install several desktop environments or a desktop and individual applications on top, UserLAnd would give each of these components its own runtime environment in its own session. This approach consumes a large amount of space, for example 850MB on the filesystem for Gimp alone. It makes more sense to opt for one desktop or one distribution and then add the

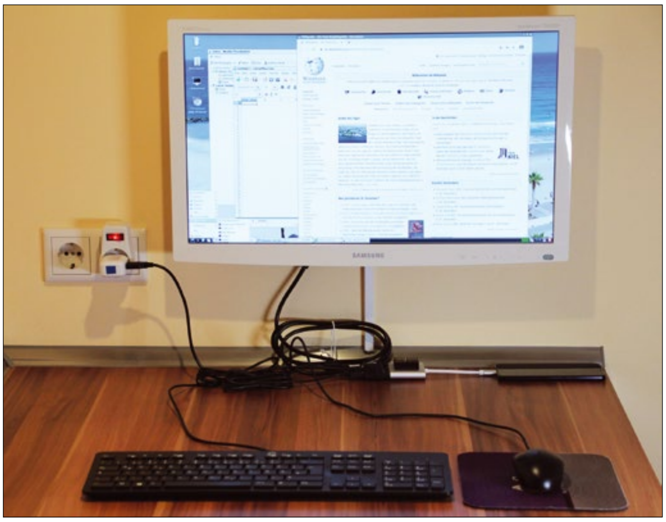


Figure 1: The smartphone is ready for use in next to no time.

Listing 1: Basic Setup

```
$ sudo apt-get update
$ sudo apt-get install vim dialog tasksel lxde
$ sudo dpkg-reconfigure tzdata
```

Listing 2: Testing Storage

```
$ ls /host-rootfs/storage
$ ls /host-rootfs/storage/emulated/0
```

Listing 3: System Expansion

```
01 $ sudo apt-get install libreoffice thunderbird
02 $ sudo apt-get install gcc g++ make
03 $ sudo apt-get install gimp vlc
04 $ sudo apt-get install xboard crafty scid fruit stockfish
05 $ sudo apt-get install cups hp-ppd hplip iputils-ping
   system-config-printer
```

Listing 4: libcrypto.so Workaround

```
$ mv /data/user/0/tech.ula/files/support/libcrypto.so.1.1 /
  data/user/0/tech.ula/files/support/libcrypto.so.1.1_orig
$ cp /usr/lib/aarch64-linux-gnu/libcrypto.so.1.1 /data/
  user/0/tech.ula/files/support
$ sudo /etc/init.d/cups start
```

applications internally, which would mean that Gimp needs only 80MB extra space.

If you decide to install a distribution and choose Ubuntu, the window manager twm first comes up with an xterm window. You can now install basic tools and the LXDE window manager (Listing 1, first two lines), which takes 20 to 40 minutes. Firefox is already included. Afterwards it makes sense to adjust the screen resolution to the monitor you are using. Edit the `~/.vncrc` file with an editor like Leafpad or vi and then restart the session. It is also important to set the time zone (Listing 1, last line).

Expansion

The Linux world is now accessible: `lsb_release -a` reports `Ubuntu 18.04.5 LTS`, and `uname -a` reveals the processor architecture – on the Snapdragon 845/855, `aarch64`. The first thing you might want to do is set the background image, the font size, or the theme.

On Android 10, the filesystem is not immediately accessible due to security policies. The feature is known as Scoped Storage and causes an app to only access app-specific directories. The UserLAnd project describes a way to import files into the Linux session [3]. It is also possible to bypass these security settings if you are aware of the risk.

To test access to the internal storage and the SD card(s), use the commands in Listing 2. In the `storage/` directory, the SD card(s) and USB sticks should appear as directories with nine-digit numbers. `emulated/0/` is usually the path to the internal storage. The DCIM directory is where the camera stores its photos and videos. You will also find other known directories, such as `Android/data/`.

You could also research manufacturer-specific paths. However, on Android 10 or higher, you can't access the paths directly: `Permission denied`. You then either have to follow the hints in the UserLAnd tutorial [3] or look at the section in this article entitled "Building Your Own UserLAnd."

System expansion continues now with applications like an office package, C, Java, Gimp, VLC, or games (Listing 3, lines 1 to 4). OpenJDK 11 JRE is already included with LibreOffice. You can set up a variety of other programs just as easily, for example Octave, Gnuplot, or Inkscape (Figure 2).

First Difficulties with Eclipse and CUPS

Installing the Eclipse IDE for Java is unexpectedly difficult; `sudo apt-get install eclipse` does not return the desired results. On the Internet, the use of Snap is recommended, but I couldn't get Snap to work either. Fortunately, you can also download the precompiled IDE from Eclipse.org [4]. What is unpleasant is that you then have to copy it manually onto the filesystem (usually to `/usr/lib/eclipse/` with a start link in `/usr/bin/`), and it does not add itself to the Start menu.

Printing with CUPS does not immediately work either. For a printer from HP, however, adding a few components is all it takes (Listing 3, line 5). In the test, printing failed at first. The cause of this failure was the `/data/user/0/tech.ula/files/support/libcrypto.so.1.1` file. I solved the problem with the workaround shown in Listing 4. Annoyingly, Android restores the `libcrypto.so.1.1` after each reboot, so the script has to run once per session. The UserLAnd project might fix this bug soon.

After you replace `libcrypto.so.1.1`, the final step is to configure `system-config-printer`. In addition, you need to install the tool again and launch it, and then you can conveniently bind (WLAN) printers. After that, the printer is available to many programs.

Sound, OpenGL, and Cameras

The UserLAnd project has a good tutorial on how to use the Android sound system [5]. You'll need to route the sound out-

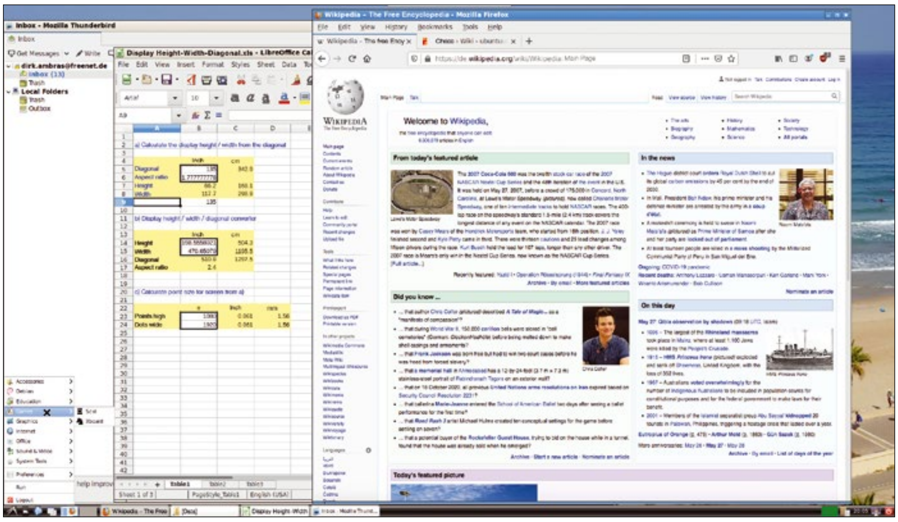

Inbox - Mozilla Thunderbird
Display Height-Width-Diagonal.xls - LibreOffice Calc
Wikipedia - The Free Encyclopedia - Mozilla Firefox

Figure 2: With little effort, you can start and run many of the standard Linux applications.

Listing 5: Additions to .bashrc

```
pulseaudio --start --exit-idle-time=-1
pacmd load-module module-native-protocol
  -tcpauth-ip-acl=127.0.0.1 auth-anonymous=1
export PULSE_SERVER="127.0.0.1"
```

Listing 6: Benchmarking with Adjustments

```
$ sudo apt-get install sysbench hardinfo
$ cd /usr/lib/aarch64-linux-gnu
$ sudo mv libc.so libc_script.so
$ sudo ln -s /lib/aarch64-linux-gnu/libc.so.6 libc.so
$ sysbench --threads=8 cpu run
$ sysbench --threads=8 memory run
$ sysbench --threads=1 --file-test-mode=seqwr fileio run
$ sysbench --threads=1 --file-test-mode=seqrd fileio prepare
[... Reboot required ...]
$ sysbench --threads=1 --file-test-mode=seqrd fileio run
```

Listing 7: Phoronix Test Suite

```
$ sudo apt-get install unzip gdebi-core
$ gdebi Downloads/phoronix-test-suite_10.0.1_all.deb
$ phoronix-test-suite run sunflow
```

Listing 8: Changes to app\gradle.build

```
compileSdkVersion 28
targetSdkVersion 28
versionName "2.7.2-Android9"
```

put via TCP to another Android process. First you have to install the Android Termux app. Add the first two lines from Listing 5 to `~/.bashrc`. Then, in UserLAnd, install the *pulseaudio* package and add the last line from Listing 5 to `.bashrc`.

Now the VLC player can also output sound – and many nice games will work too (see the box entitled "Games"). However, some games only display an error message because I have not yet been able to install OpenGL. I also couldn't get the Linux system to access the smartphone's cameras.

Performance

Once you get Linux up and running on your Android phone, the next question might be, does it provide the performance you'll need to make it a viable option? You can test the performance with exactly the same benchmarks you would use on a real PC to compare the results directly. Sysbench requires small adjustments (see Listing 6).

The Hardinfo GUI tool depends on the VNC viewer. RealVNC is clearly the winner with 1,449 points compared with bVNC which only scores 930 points. Before running the `seqrd fileio run` test, you need to restart the Linux system to avoid unrealistic results due to caching. The Phoronix test suite [6] (Listing 7) also provides interesting results. I would also have liked to run the 3D graphics tests like `unigine-heaven` from this suite, but they are only available in binary format for x86_64 machines.

The surprise: An average family computer clearly outclasses a smartphone, but cannot totally outrun it. The 3D graphics performance is probably the only real drawback (Figure 4).

Alternative Browsers

During the tests, Firefox did not run at times, for example, due to problems with fonts. You can install a different browser if you are having trouble with Firefox [7]. Not all of browsers are available for the ARM architecture, but Chromium and Epiphany work. For a successful start, you have to add the option `--no-sandbox`.

Building Your Own UserLAnd

Why compile UserLAnd yourself? First, compiling UserLAnd lets you work around the Android 10 storage policy. Second, you might have new ideas and want to collaborate on the project. If the motive is to disable the storage restrictions, you should definitely be aware of the risk of removing what is in principle a sensible security barrier.

You can download the source code of UserLAnd directly from GitHub [2] and then open it with Android Studio 4.x [8]. If you want to bypass the new storage policy, you can compile UserLAnd with an older Android SDK, such as version 28, which predates the restrictions. To install version 28, you have to edit the `app\gradle.build` file in three places (Listing 8).

As usual, you'll need to set your own device as a developer device and install the software APK from the PC with adb [9] or via the file manager in the device. Annoyingly, you have to uninstall the official UserLAnd version beforehand (due to the different or missing signature) and thus lose all data the Linux system set up.

Limits and Problems

Probably the most annoying weakness of the UserLAnd solution for Linux on Android is the limited number of open windows (three to six), which can even decrease during the session. UserLAnd then reports

```
Maximum number of clients reached. Failed to open display.
```

What is also very annoying is that the connection to the VNC server occasionally breaks down. Then you lose all your unsaved data because you can't connect to the session anymore.

Games

I could only check a few of the large number of games available in Ubuntu. Some refuse to launch completely unless you set up sound output or OpenGL first. However, some games do run and are easy to install, including Pingus, Maelstrom, and Gnome Sudoku, as well as XBoard with the chess engines Crafty and Fruit (Figure 3).

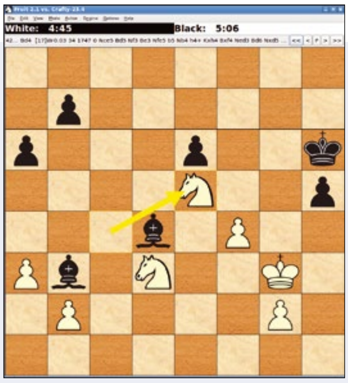


Figure 3: Crafty and Fruit are strong chess engines, playing against each other here on XBoard.

Some VNC viewers had trouble transferring keystrokes for non-English character sets. AndroidVNC does not have the problem, but it is no longer maintained and has many other shortcomings. RealVNC's PC viewer, on the other hand, transfers the key combinations correctly.

Wireless mice often react clumsily. This seems to be a hardware problem that cannot be solved via the energy-saving settings. It is also annoying that you cannot shut down a Linux session and have to stop it in Android instead.

On the positive side, the filesystem can be exported and used as a backup – very handy for transfers to other devices, but keep in mind that backing up the filesystem can take some time.

Conclusions

The age of full-fledged pocket-sized computers has arrived. Once a few teething problems have been solved, these computers can be used at home without restrictions. Smartphones can already handle many basic applications, but you still need modern desktop computers or laptops for special tasks and 3D games with high performance and resource requirements. I also own a smartwatch with a dual-core CPU and 4GB RAM, and I am already wondering whether I'll be able to get a Linux distribution running on it. ■■■

Info

[1] Android source code: *https://source.android.com/*

[2] UserLAnd: *https://github.com/CypherpunkArmory/UserLAnd*

[3] Importing files: *https://github.com/CypherpunkArmory/UserLAnd/wiki/Importing-and-exporting-files-in-UserLAnd*

[4] Eclipse downloads: *https://www.eclipse.org/downloads/download.php?file=/eclipse/downloads/drops4/R-4.17-202009021800/eclipse-SDK-4.17-linux-gtk-aarch64.tar.gz*

[5] Using the Android sound system: *https://github.com/CypherpunkArmory/UserLAnd/issues/371*

[6] Phoronix test suite: *https://www.phoronix-test-suite.com/?k=downloads*

[7] Alternative web browsers: *https://www.ubuntupit.com/13-best-ubuntu-web-browsers-which-one-is-right-for-you/*

[8] Android Studio: *https://developer.android.com/studio*

[9] adb: *https://developer.android.com/studio/command-line/adb*

Author

Dirk Ambras works at Airbus as an IT expert. He used to work for Siemens as an operating system developer for cell phones in the areas of graphics, games, and Java. The fun of working with embedded devices is something he has never lost.

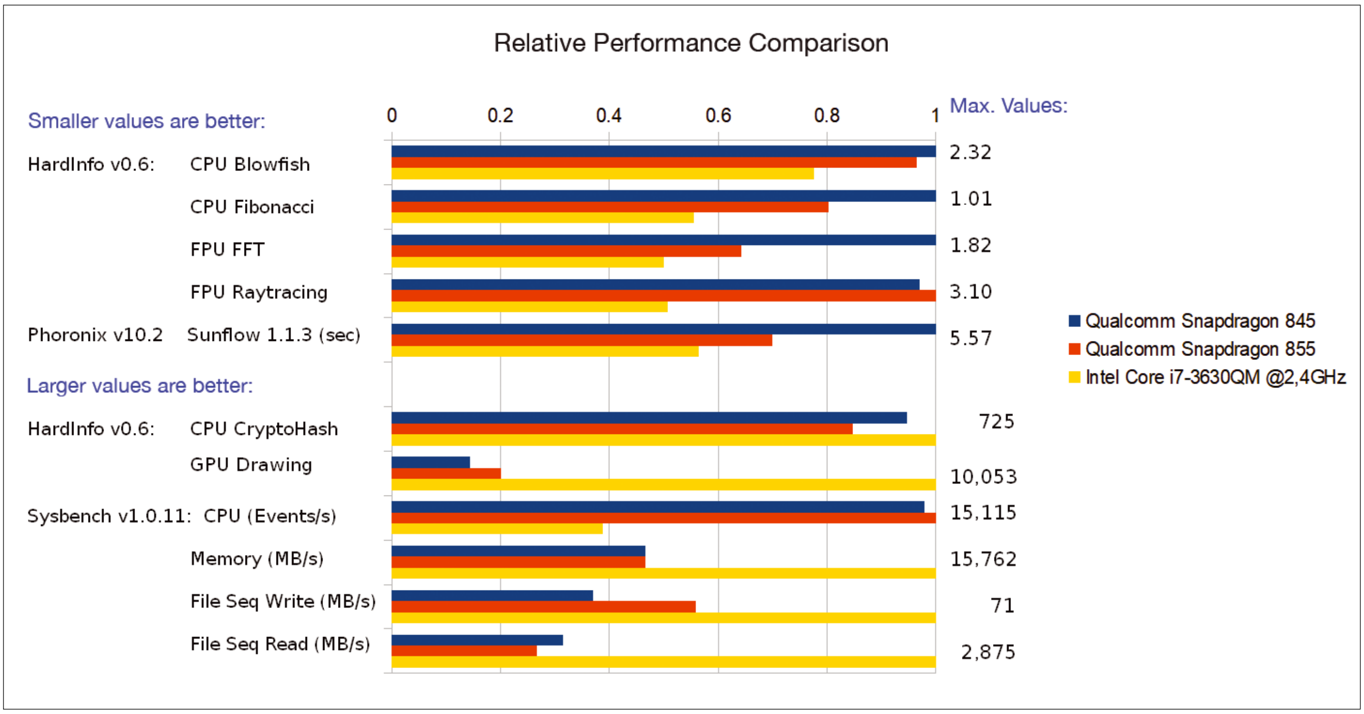
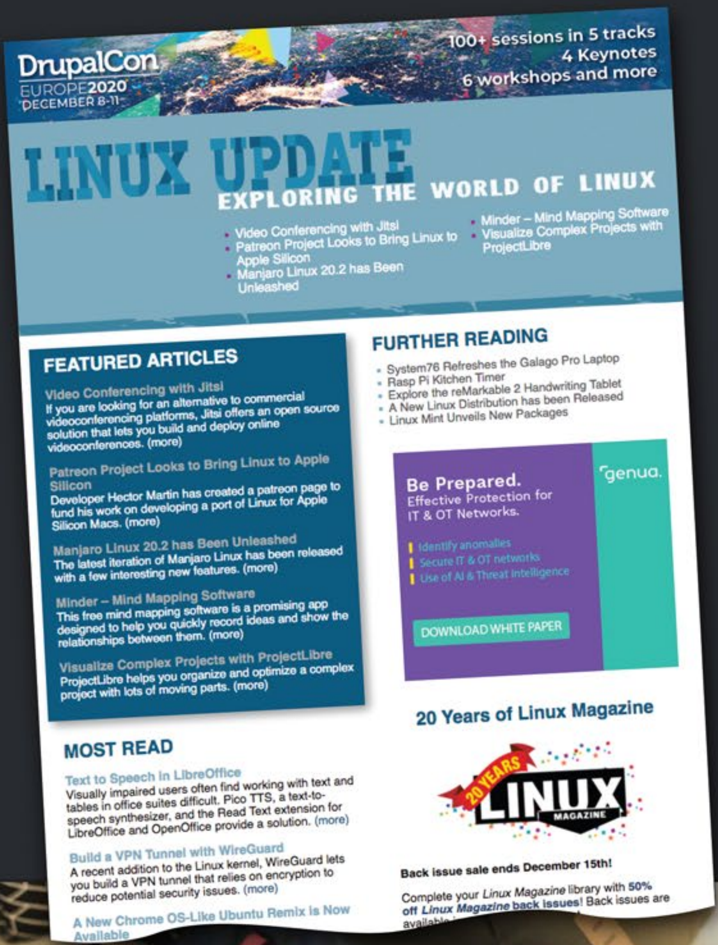


Figure 4: Smartphones can keep up with PCs for some tasks. The performance is sufficient for general office work, web browsing, and certain games that aren't too demanding.



DrupalCon
EUROPE2020
DECEMBER 8-11

100+ sessions in 5 tracks
4 Keynotes
6 workshops and more

LINUX UPDATE
EXPLORING THE WORLD OF LINUX

- Video Conferencing with Jitsi
- Patreon Project Looks to Bring Linux to Apple Silicon
- Manjaro Linux 20.2 has Been Unleashed
- Minder – Mind Mapping Software
- Visualize Complex Projects with ProjectLibre

FEATURED ARTICLES

Video Conferencing with Jitsi
If you are looking for an alternative to commercial videoconferencing platforms, Jitsi offers an open source solution that lets you build and deploy online videoconferences. (more)

Patreon Project Looks to Bring Linux to Apple Silicon
Developer Hector Martin has created a patreon page to fund his work on developing a port of Linux for Apple Silicon Macs. (more)

Manjaro Linux 20.2 has Been Unleashed
The latest iteration of Manjaro Linux has been released with a few interesting new features. (more)

Minder – Mind Mapping Software
This free mind mapping software is a promising app designed to help you quickly record ideas and show the relationships between them. (more)

Visualize Complex Projects with ProjectLibre
ProjectLibre helps you organize and optimize a complex project with lots of moving parts. (more)

FURTHER READING

- System76 Refreshes the Galago Pro Laptop
- Rasp Pi Kitchen Timer
- Explore the reMarkable 2 Handwriting Tablet
- A New Linux Distribution has been Released
- Linux Mint Unveils New Packages

genua.
Be Prepared.
Effective Protection for IT & OT Networks.
- Identify anomalies
- Secure IT & OT networks
- Use of AI & Threat Intelligence

DOWNLOAD WHITE PAPER

MOST READ

Text to Speech in LibreOffice
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. (more)

Build a VPN Tunnel with WireGuard
A recent addition to the Linux kernel, WireGuard lets you build a VPN tunnel that relies on encryption to reduce potential security issues. (more)

A New Chrome OS-Like Ubuntu Remix is Now Available

20 Years of Linux Magazine

20 YEARS LINUX MAGAZINE

Back issue sale ends December 15th!

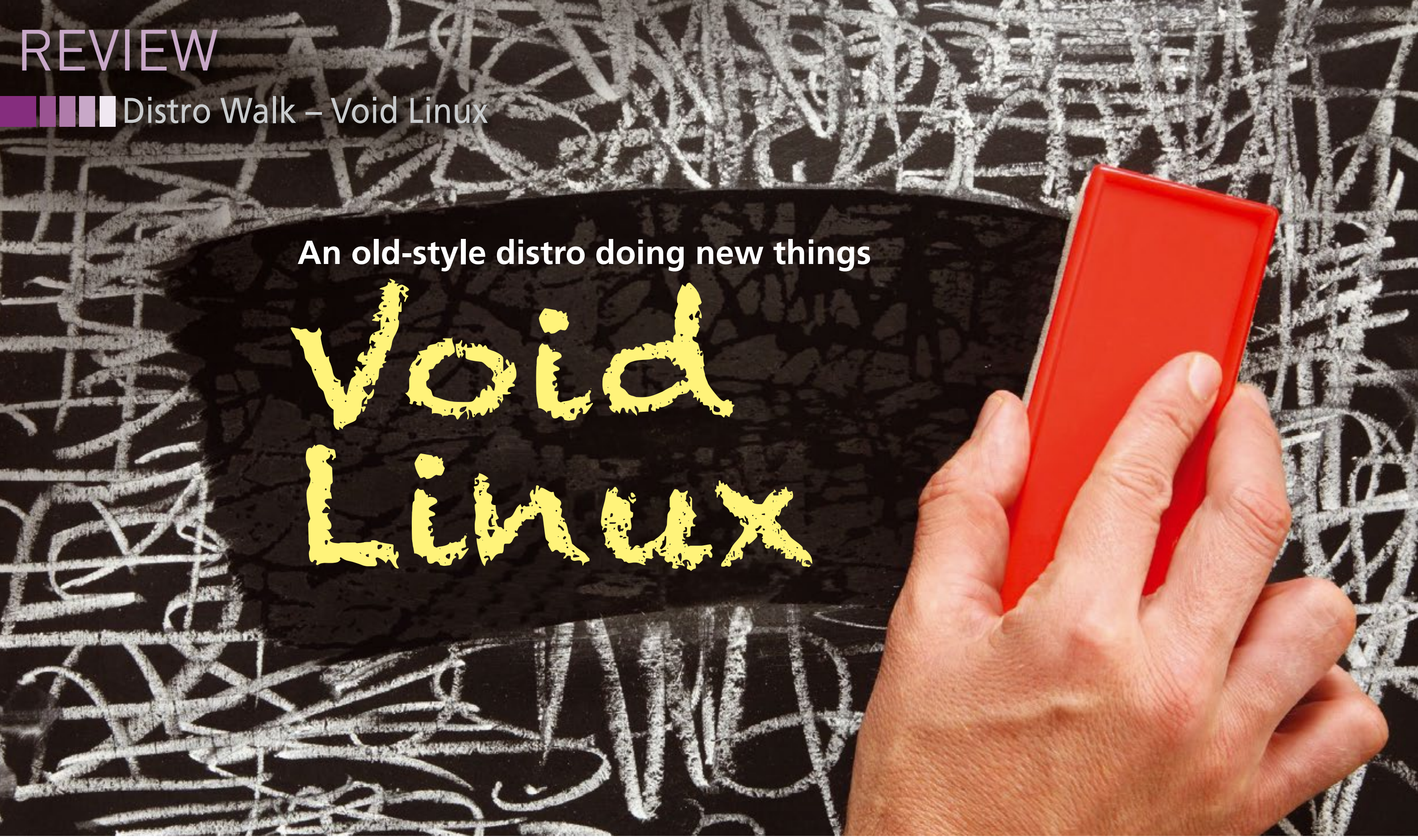
Complete your *Linux Magazine* library with **50% off *Linux Magazine* back issues**! Back issues are availabl

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**

bit.ly/Linux-Update



An old-style distro doing new things

Void Linux

Void Linux offers a unique distribution with a lower barrier to participation that is easy to manage.

By Bruce Byfield

DistroWatch lists 278 active distributions, but these numbers are misleading. Many distributions are minor variations of a half dozen major distributions, distinguished by their default software selection or oriented to a particular task. A notable exception is Void Linux [1], a small project organized more like the distributions of two decades ago, with every part of its structure carefully considered. The result is one of the more original distributions available today. To learn more, I contacted Michael Aldridge, who answered my questions after consulting with other core members of the development team.

Originally released in 2008 by Juan Romero Pardines, Void Linux served as a testbed for the XBPS Package Manager. Since then, Void has changed direction several times. Currently, Void Linux is a rolling release distribution with an emphasis on making system management and contribution easy.

Aldridge describes Void as "a barely controlled anarchy" with "quasi-appointed leads spread across infrastructure, platform support, the package manager itself, and the creature comforts such as docs and debugging tools …. Because we have so few members, and all of them are volunteering their time, we simply cannot spare the clock cycles to have a stricter organization." Developers tend to control what interests them, while every core maintainer has the right to vote on decisions about the project as a whole, such as accepting new members.

Currently, Void has 17 core maintainers. Since January 1, 2021, 266 maintainers have contributed to the distribution's packages and 16 to the official handbook. The project is run on minimal hardware: three build servers, plus another seven servers and a handful of mirrors. Aldridge says, "We rely on the generosity of other projects and organizations to mirror packages to end users."

Over 8,000 packages for nine supported architectures are officially maintained, and additional ports are maintained unofficially by project members. "As a project that highly respects the privacy of our end users," Aldridge says, "we don't track metrics across any large pool, in terms of ISO downloads or mirror usage." However, Void's default mirror delivers an average of 38TB of data each month, a testimony to the distribution's growing popularity.

Design Philosophy

Void's name was chosen for lack of a name and stands for nothing. In the same way, Aldridge notes that "Void's lack of philosophy is a philosophy in itself." Nonetheless, several general principles have evolved over the years. First, the only changes to upstream applications are those needed to compile Void, just as in Arch Linux. As much as possible, configuration is left to the user. "This approach has led many users to label Void as minimalistic," Aldridge notes, "but our simplicity comes from Void trying to not interfere in upstream's decisions, not a drive for minimalism."

Second, Void Linux discourages nonstandard templates for packages, instead encouraging a standard build logic. "This means," Aldridge explains, "that tooling changes, compiler flags, new lints, and any other changes are immediately applied to all newly built packages. Furthermore, wide reaching changes can be coordinated within a single repository, and we don't have to keep our tooling compatible with old style templates if there changes. Anybody who maintains their own private templates needs to rebase [using] our work, because our repository is the source of truth at any given commit."

Lead Image © Stephen Rees, 123rf.com

This insistence on consistency makes packaging easier for developers and makes consistency in the distribution easier to maintain.

This consistency is also partly responsible for Void's third design principle: the ability to maintain what its website calls a "stable rolling release." As Aldridge points out, whether this description is a contradiction depends on the definition of "stable." In a number of distributions, a rolling release means a high risk of problems when updating. By contrast, Void uses a number of mechanisms to avoid such problems. For example, its build system has checks for shared library compatibility, which minimize the chances of partial updates. Similarly, `xbps-src`, its build system, includes a large number of sanity checks. In addition, when a kernel is updated, the older one is not immediately deleted, which ensures that the system remains bootable if the new kernel has problems. Users can also use a tool called `vkpurge`, which allows a controlled removal of old kernels and a choice of current kernels for replacement, including long-term support kernels. Another tactic to maintain a stable rolling release is to use only stable upstream releases in most cases. "Admittedly, it's a moving target," Aldridge says. "We heavily rely on what the upstream considers stable, and there will be always issues and bugs slipping through. But, again, rolling releases give us a tool to quickly fix those issues as they occur."

Distinguishing Features

Besides the features required for a rolling release, Void has several distinguishing features. Its default installation is minimal, and much of the configuration is left to users. In addition, unlike most distributions, all the supported architectures are built from a single source, using the XBPS Package Manager as a wrapper for build functions. "No matter if you choose glibc on x86_64 or ARMv7 with the musl libc," Aldridge says, "the very same package definitions are used. This is true for all our platforms except our ports, such as PowerPC platforms, which are currently developed in a fork that regularly is synchronized with the Void Linux project, but we strive to include it in the main project. Our software supports several architectures (MIPS, ARMv5) [that] we as a project do not officially support – mostly because computing resources are scarce, and we need to decide how to spend them."

Another unusual feature is Void's C library diversity: the support of multiple libraries such as musl libc and glibc in the same system. Originally, this support was included simply because it was possible, according to Aldridge. Then the development team realized that each library had its own advantages. For instance, it turned out that using musl libc made it easier to explore the use of glibc. "It was just the logical conclusion that if we rely on musl for development, why not make it a target for the distribution?" Aldridge explains. "With our approach, we give ourselves and users the choice: rely on a very clean and readable solution like musl libc or opt for a more widely used solution like glibc, which contains more platform-specific optimizations and is what most upstreams use to test their software. This flexibility makes it easier to run closed source, third-party software."

Probably, however, Void is best known for using runit as an init system. Void was among the first distributions to run systemd, but the developers quickly found that it had portability issues and that upstream projects largely did not support Void's alternate C library implementations. Consequently, Void has used runit since 2014. Runit is not feature complete, because it lacks support for features such as oneshot services, but on the whole, Void is simple enough that runit meets most of its needs. "We've been keeping an eye out for init systems that fit our needs," Aldridge says, "but … a new system would have to meet our requirements in simplicity and not be a step backwards into SysVinit-like systems. All in all, because of the simplicity of the concept, the service scripts are easy to read, easy to understand, and given runit's small feature space, easy to write. [Runit] somehow came to fit our ethos, even if we only adopted it because of portability issues."

What's Next

Aldridge says, "We are very satisfied with the current state of both the project and the system." However, the expansion of multiple library support and improvements to the XBPS Package Manager remain ongoing goals. In addition, Void is considering expanding the use of containers on its services to make the introduction of new hosts easier and to allow machines to be pulled out of service for updates and maintenance. Void Linux is also considering the introduction of other hardware architectures into the distribution.

As for other features, who knows? By going its own way, Void has produced not only an original distribution, but an efficient one – and one that makes contributing relatively easy. Sometimes publicity – such as from participating in Hacktoberfest the past few years – results in a patch load that strains the project's maintainers, but the low barrier to participation remains a point of pride. "We enjoy being a serious project where some people make their very first FOSS contributions," says Aldridge. ■■■

Info

[1] Void Linux: *https://voidlinux.org/*

Photo by Samule Sun on Unsplash

Tools for converting multimedia files

Transformers

Multimedia files are easy to convert – if you have the right tools. This month we look at some useful applications that convert video and audio files. *By Erik Bärwaldt*

Tutorials, screencasts, or even training videos are becoming increasingly important in everyday life. Multimedia is particularly useful in the home office and for homeschooling. But not every device can handle every file format, and not every screen displays all resolutions. Conversion programs help to adjust the formats and modify the image resolutions for the best possible results. Linux offers a great variety of multimedia conversion tools. We decided to investigate a few of the top contenders. Note that we focused on active projects for this report and did not consider some of the alternatives that are still widely known but not in active development (see the box entitled “Out of the Running”).

Out of the Running

Numerous graphical front ends for multimedia back ends are available on the Internet. However, many of these tools have not been maintained or developed for years. For example, Transcoder has been orphaned since 2011, and OggConvert has not seen an update since 2010. Other software, such as FFmpegYAG, can no longer be installed on current distributions due to unresolvable dependencies.

Containerized

Audiovisual content can be stored with different parameters and image resolutions. Audio files can be saved in lossless or lossy formats with different bit rates. Container formats, such as MP4, MKV, and AVI are popular for video files. A single container can hold multiple files. For instance, a movie container can store several audio files in different languages. Since the codecs vary, the container format alone does not allow any conclusions about the quality of the contents.

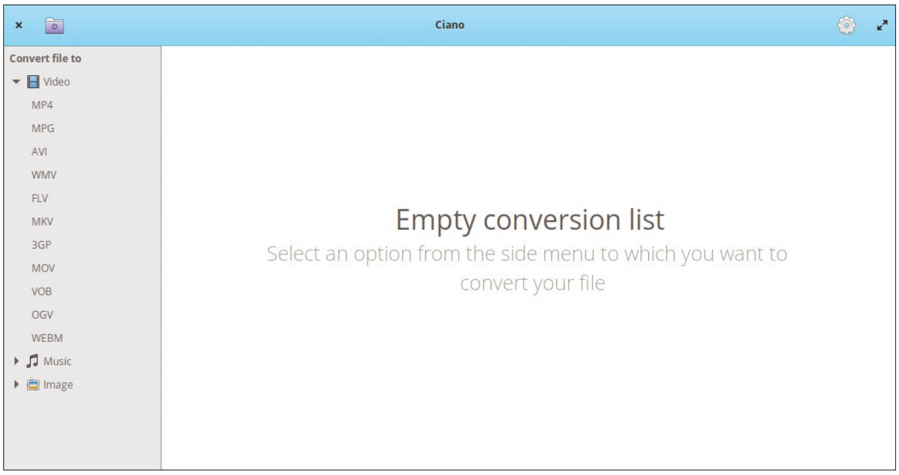
Linux supports most free multimedia formats and codecs out of the box. Commercial packages also make proprietary codecs available; Linux can clearly score points in direct comparison with other operating systems as a platform for converting and playing multimedia content.

Existing solutions for converting multimedia content offer some flexibility: Often you can modify the resolution or bit rate to adapt to external playback options. For this purpose, most Linux applications for converting multimedia content rely on one of the leading open source multimedia frameworks. Many of the tools use FFmpeg [1], but GStreamer [2] also forms the basis for some conversion programs.

Ciano

Ciano [3] is a multimedia conversion tool that keeps with the philosophy of providing the easiest possible user interface. The Ciano conversion utility has an easy program interface that requires no training.

Ciano uses FFmpeg as its back end. Because Ciano can also convert image files, you’ll need to install ImageMagick on the system. Several Linux distributions have added Ciano to their package archives, and you will also find a deb package on the project website.



Empty conversion list
Select an option from the side menu to which you want to convert your file

Figure 1: Ciano requires no training thanks to a very simple structure.

Ciano's interface (Figure 1) is divided into three parts: On the left of the window is a vertical bar with the different target formats. To the right is the large work area, where you can see a list of the files to be processed. A small settings dialog resides in the titlebar, which you can access via a gear icon.

To convert content, click on the desired target format in the format bar on the left. In the dialog that opens, click on the + icon and then select the desired files in the file manager. Ciano will transfer the files to the list view. Click on the *Start conversion* button to start the conversion.

The selection window then closes. Ciano loads the contents into the working area of the main window and starts the conversion. A bar below each file indicates the progress of the action. Ciano can convert several files simultaneously.

Curlew

Curlew [4] is a graphical user interface for converting audio and video files written in Python and based on the FFmpeg framework. The still-quite-young project is available in the repositories of many distributions. In addition, you will find the software as a tarball on SourceForge. Installing from the repositories creates a separate launcher in the menu hierarchy.

Curlew opens a clearly arranged program window with a state-of-the-art look. In the usual style for the Gnome desktop and its applications, Curlew combines the most important control elements in the titlebar (Figure 2). An initially empty table view takes up most of the window. It later fills up with a list of selectable files, which also contains information about the file size, as well as time information for the conversion.

At the very bottom of the window, you will find a button that covers almost the entire width of the window, which you can use to select the target format. Clicking on the button opens the list of target formats in a separate window. In addition to the conventional container formats, the list contains numerous hardware-specific formats. Double-click one of the listed formats to add it to the favorites list.

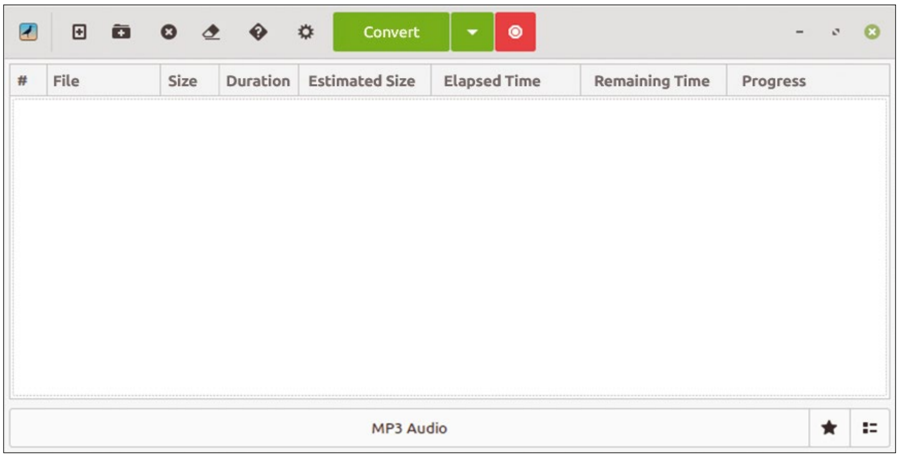


Figure 2: Curlew impresses with a very easy-to-use interface.

You can configure individual settings for each of the listed formats. Clicking on the gear icon in the titlebar opens a dialog in the list area of the program window with a tab structure that supports format-dependent modifications.

The settings dialog also lets you click to configure a two-pass setting, where the second conversion run is supposed to offer better image quality. Checking another box tells the application to convert only the video track but not the audio track.

Once you have configured all the settings, click the gear icon in the titlebar to go to the empty workspace. If necessary, save the configuration up front as a favorite so that you can use the settings in later conversion runs.

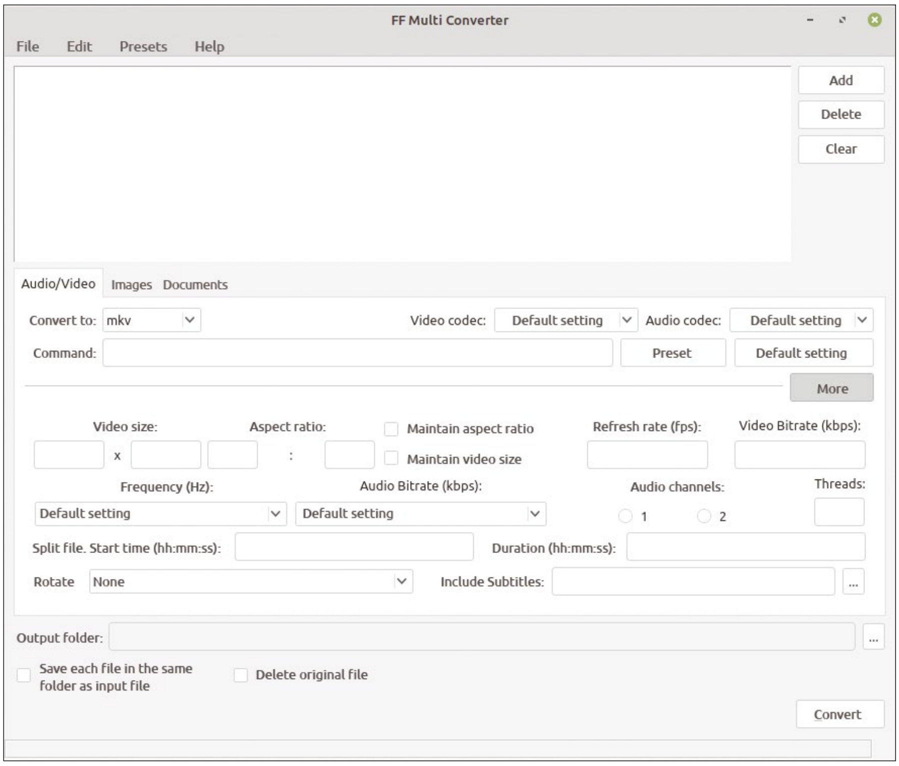
To convert multimedia content, drag and drop the source file from a file manager into the workspace of the program window. Curlew lists the individual files one below the other. After you click the *Convert* button top center in the window, the software starts to convert the source file.

Converting high-resolution video files takes a long time even on state-of-the-art computer systems. If you want to stop the conversion process, click the red *Stop* button. To view the contents of a file in the conversion list, right-click on the desired entry and select *Play* in the context menu. Curlew now opens the preset playback software and plays back the content. You can even play back a file that the program is currently converting.

FF Multi Converter

FF Multi Converter [5], which is also based on FFmpeg, converts documents and image files as well as multimedia files. On the back end, ImageMagick converts the graphic files and Universal Office Converter (unoconv) converts the documents. The software, written in Python and based on the Qt libraries, is available in the repositories of most common distributions.

The FF Multi Converter main window appears somewhat confusing (Figure 3). In its upper third is a list area where the source files you wish to convert are loaded with the help of a small file manager. Below that, you can configure the conversion in a settings dialog. The



Convert to: mkv
Video codec: Default setting
Audio codec: Default setting
Command: Default setting
Frequency (Hz): Default setting
Audio Bitrate (kbps): Default setting
Rotate: None

Figure 3: FF Multi Converter opens a window with many options.

settings are divided into the three tabs *Audio/Video*, *Documents*, and *Images*.

For video files, use the preset dialog to specify the container format in the *Convert to:* selection box. Next to it, you will find more selection boxes, where you can choose the video and audio codecs. If you click on the *More* button, you'll find an extended settings area that lets you configure additional parameters for the target file. For example, you can set the output quality and modify the aspect ratio of the target file. If you have particularly powerful hardware, specify the number of processes to run simultaneously in the *Threads* field.

At the bottom of the window, enter the output path and check a box to define whether the software deletes the source file after conversion.

When you are finished configuring settings, press the *Convert* button bottom right. The software then opens a dialog with a progress bar. Pressing the *Details* button also opens a virtual terminal display within the window, in which the program continuously outputs status messages.

Simultaneous playback for the content you are converting is not available, but if you need this option, you can always turn to external playback software.

FFQueue

FFQueue [6] is available on the project's website as a generic `tar.xz` archive for distributions with DEB package management. You will also find a link to the source code, which is licensed under the GPLv3. If you are using Debian, Ubuntu, or one of their many derivatives, simply unpack the `tar.xz` archive and then grant execution rights to the *FFQueue* binary by running the command:

```
chmod +x FFQueue
```

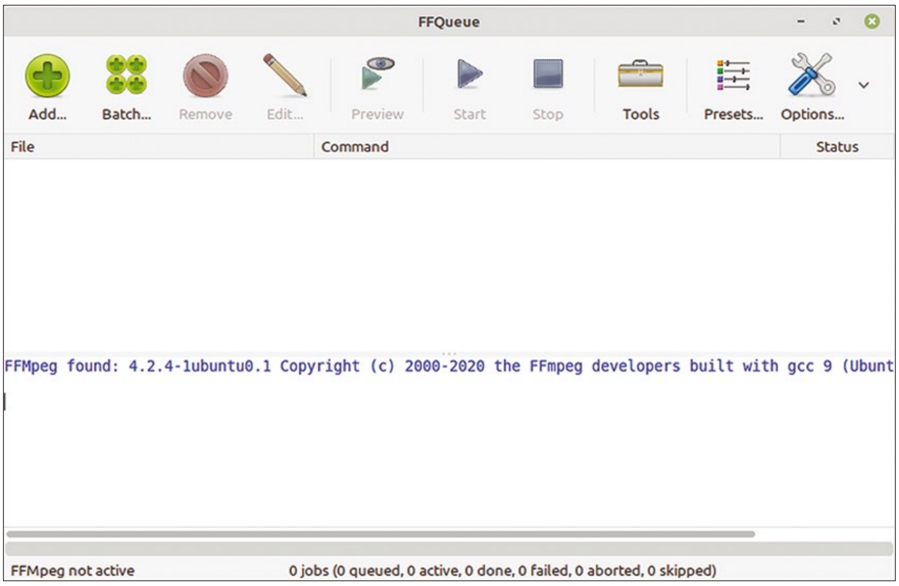
You can then start the software with the `./FFQueue` command.

The surprisingly fast program then opens a somewhat unconventional main window (Figure 4). Instead of the usual menus, you will find a buttonbar at the top with two work areas arranged below it. The files you wish to convert appear in the upper segment. Below them, FFQueue displays real-time notifications during transcoding, which it also records in its logfile.

FFQueue does not offer a full-featured settings dialog but expects some system management skills on the part of the user. Since the software relies on FFmpeg as its back end, the configuration dialog offers the option to define the search path for the FFmpeg binaries individually.

You can also specify which external media player you want to use for previews. FFQueue explicitly proposes the VLC player. If audio tracks or subtitles are available in separate files, specify the file extensions in the settings dialog. If you want to pass additional special parameters to FFmpeg, enter the parameters in the *Custom console command* input field.

You create settings profiles in the *Presets* dialog, which you can open by pressing the button of the same name. The settings contain detailed presets for audio and video codecs, as well as the general parameters. After opening the



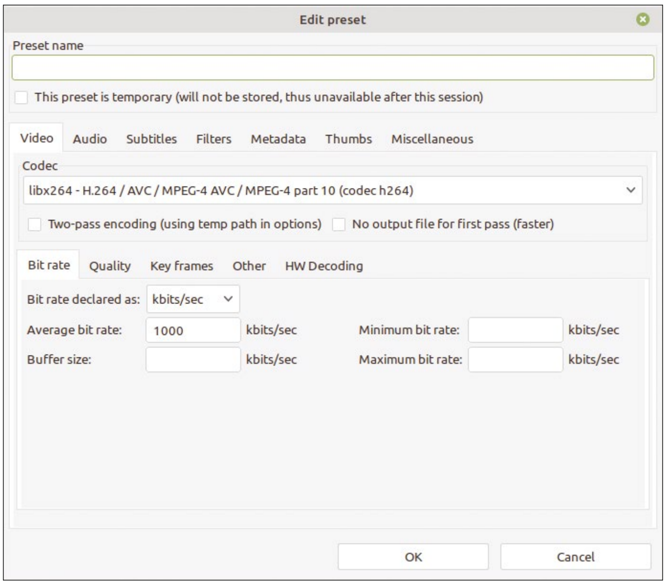
```
FFmpeg found: 4.2.4-1ubuntu0.1 Copyright (c) 2000-2020 the FFmpeg developers built with gcc 9 (Ubunt
```

FFmpeg not active
0 jobs (0 queued, 0 active, 0 done, 0 failed, 0 aborted, 0 skipped)

Figure 4: FFQueue provides all the necessary information in the program window.

Preset manager dialog, click the *New* button to open a window where you can create a profile (Figure 5). Enter the name of the new profile at the top and then make the appropriate settings in several tabs. Once you have set everything up as desired, press the *OK* button bottom right to transfer the new profile to the profile manager.

You then have two possibilities for starting the content conversion. The *Add* button in the upper left corner of the main window opens a file manager in which you select the file to be converted. Alternatively, you can tell FFQueue to convert several files in batch mode. To use batch mode, you must have at least one settings profile in the *Presets* dialog.



Edit preset

Preset name:

□ This preset is temporary (will not be stored, thus unavailable after this session)

Codec: libx264 - H.264 / AVC / MPEG-4 AVC / MPEG-4 part 10 (codec h264)

□ Two-pass encoding (using temp path in options) □ No output file for first pass (faster)

Bit rate declared as: kbits/sec

Average bit rate: 1000 kbits/sec   Minimum bit rate: kbits/sec

Buffer size: kbits/sec   Maximum bit rate: kbits/sec

Figure 5: Using profiles to manage settings in FFQueue.

Once you have created a profile, press the *Batch* button in the main window to open the file manager and use Ctrl + click to select content. Then click on *Open* to open the batch editor. The batch editor enables granular configuration of the conversion process.

The *Dry run…* button lets you check whether your configurations for the batch jobs are correct. The software tests the job configuration and displays the matching results at the bottom of the main window. If FFQueue indicates correct job settings, you can start the individual jobs for batch processing by pressing the *Make jobs…* button. The program lists the jobs it created individually at the bottom of the main window.

Press the *Start* button to start processing the individual jobs.

HandBrake

HandBrake [7] has firmly established itself in the Linux universe as a multifunctional program for transcoding multimedia content. All of the popular Linux distributions have prebuilt versions of HandBrake in their software repositories, and you will even find even a package for the command line.

HandBrake, which is based on the FFmpeg framework and GTK + , transcodes optical media and comes with some predefined profiles that contain settings for special end devices, such as Sony PlayStation and Apple's iPod. You will also find profiles optimized for platforms like Vimeo and YouTube. In batch mode, HandBrake processes several jobs in succession if required.

HandBrake supports many current codecs. For example, you can convert visual content to H.264 or H.265, but also to MPEG-2 and MPEG-4, as well as VP8 and VP9. In addition, HandBrake supports the Theora codec. For audio tracks, the application can handle AAC, MP3, Ogg Vorbis, AC3, and FLAC (16- and 24-bit).

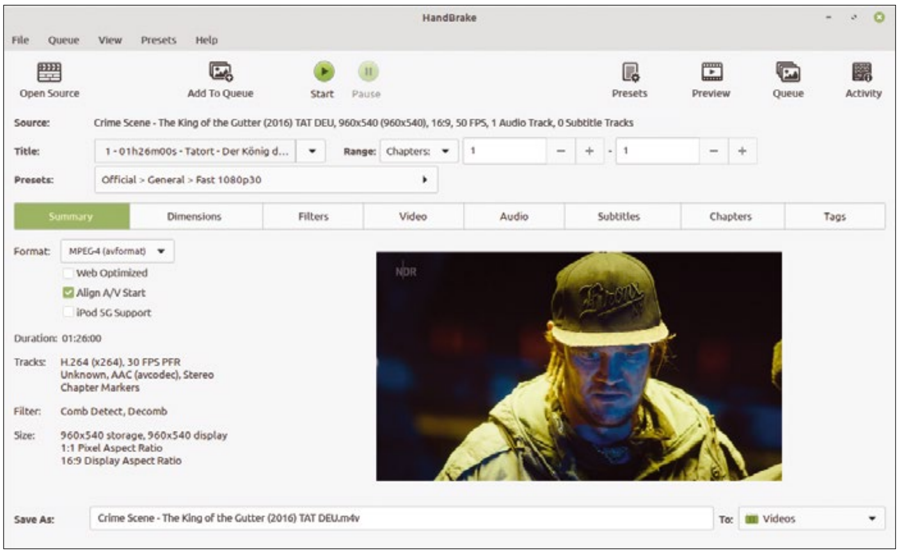
The interface in the graphical version of HandBrake (Figure 6) appears somewhat confusing for the uninitiated. In addition to a menubar and a buttonbar below, you will find setting options for the transcoding job in the work area.

Specify the source file using the *Open Source* button top left in the window. If the video is in MP4 or MKV format, you will only see one chapter; if the video is a DVD or Blu-ray disc, you will see the number of individual chapters. A preview image of the loaded source video appears in the program window.

Below the chapter and profile overview, you will find the actual workspace. HandBrake lists the format, codecs used, resolution, and aspect ratio of the current source file. You can set other important options in the *Video*, *Audio*, *Subtitles*, and *Tags* sections.

In the *Video* section, you can configure specifications for the frame rate, encoder, bit rate, and image quality. In *Audio*, you will find the associated options for the audio track. If you have loaded a video from an optical disc with several audio tracks for language variants, you can integrate the desired audio tracks into the target file in this dialog. Professional playback software, such as the VLC media player, detects the additional tracks and offers a menu in which you select the track.

The same applies to the *Subtitles* tab, where you integrate any existing subtitle tracks into the new target file. By default, subtitles are not displayed automatically,



Source: Crime Scene - The King of the Gutter (2016) TAT DEU, 960x540 (960x540), 16:9, 50 FPS, 1 Audio Track, 0 Subtitle Tracks

Title: 1 - 01h26m00s - Tatort - Der König d…   Range: Chapters: 1 - 1

Presets: Official > General > Fast 1080p30

Format: MPEG-4 (avformat)

□ Web Optimized
☑ Align A/V Start
□ iPod 5G Support

Duration: 01:26:00

Tracks: H.264 (x264), 30 FPS PFR
Unknown, AAC (avcodec), Stereo
Chapter Markers

Filter: Comb Detect, Decomb

Size: 960x540 storage, 960x540 display
1:1 Pixel Aspect Ratio
16:9 Display Aspect Ratio

Save As: Crime Scene - The King of the Gutter (2016) TAT DEU.m4v   To: Videos

Figure 6: HandBrake shows all the settings for the source and destination files in a tab structure.

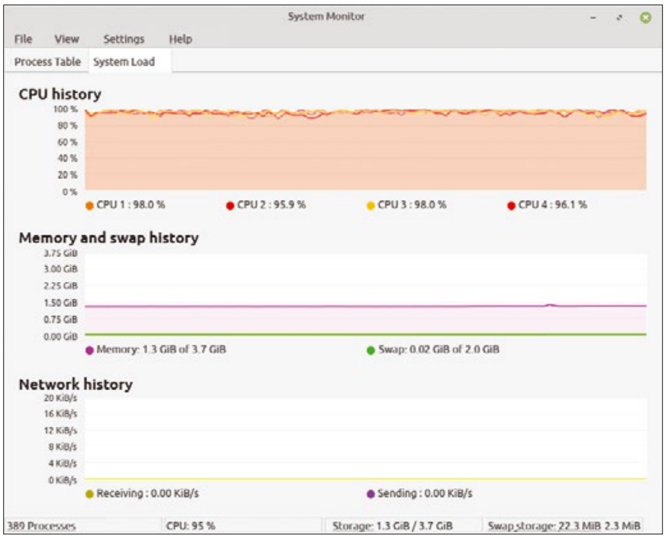


Figure 7: HandBrake generates a heavy load on the hardware.

but can choose to display subtitles in a drop-down menu. HandBrake saves the subtitle tracks directly in the target file.

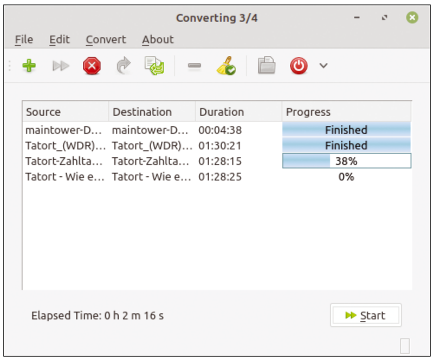
The *Tags* tab lets you tag the file with keywords to make it easier to categorize and index later on. For this purpose, HandBrake offers several fields for free text input, where you can enter, say, the director or actor and also the genre.

Once you have made all the settings, press the *Start* button top center in the program window's buttonbar. The software will the start transcoding,

HandBrake is not exactly frugal with resources during transcoding. On conventional desktop computers with quad-core processors, the application generates heavy load on all the CPU cores, even if the CPU supports hyperthreading (Figure 7).

QWinFF

The small QWinFF [8] program acts as a graphical front end for FFmpeg and relies on the Qt libraries. The software is available in the repositories of almost all distributions. QWinFF shows an extremely spartan program window after installation and startup. In the window, the large list area shows the files to be converted. Above this area is a buttonbar and a small menubar.



Source	Destination	Duration	Progress
maintower-D...	maintower-D...	00:04:38	Finished
Tatort_(WDR)...	Tatort_(WDR)...	01:30:21	Finished
Tatort-Zahlta...	Tatort-Zahlta...	01:28:15	38%
Tatort - Wie e...	Tatort - Wie e...	01:28:25	0%

Elapsed Time: 0 h 2 m 16 s

Figure 8: QWinFF processing several jobs one after the other.

To convert a file to another format, just drag it from a file manager and drop it into the program window. QWinFF then opens a new window in which the file appears in a table view. To select additional content for conversion, press the plus button to the right of the table view and select the other files from the file manager.

Because the table view includes a wizard, you can set all configuration options for the conversion in a few predefined steps. First select the content, and then click on the *Next* button. A dialog lets you specify the target formats and the output path.

QWinFF is based on FFmpeg, so you will find the entire range of codecs supported by FFmpeg and installed on the system in the *Convert to* selection field. Caution: QWinFF does not distinguish between audio and video codecs. If you accidentally set the wrong output format, the video or audio track might be disabled.

The software also comes with profiles for common use cases, which you select in the *Preset* selection box. Use the *Edit* button to modify a profile if necessary. In the *Advanced* tab, you then enter individual parameters of the FFmpeg command set directly in an input window.

To set detailed options for the output files, select a file you wish to convert and right-click to open a context menu. The *Set Parameters* entry takes you to the Conversion Parameters dialog, where you can modify parameters for the video and audio codecs and the image resolution. Checking the box lets you hide the video or audio track completely.

After completing all the settings, press *OK* to confirm your selection; then press *Start* to start the conversion. QWinFF processes the files listed in the table one after the other (Figure 8). The conversion does require some hardware resources, but at least the computer remains operable.

Conclusions

Thanks to the solid multimedia foundation provided with Linux, you have several reliable options for converting video and audio content. The graphical front ends remove the need to research and memorize command line parameters and provide a useful overview of individual configuration settings.

Users with professional needs who insist on granular settings are best served with HandBrake, but the many HandBrake options add to the learning curve. On the other hand, if you are hoping to start converting files immediately without background knowledge, you are better off with QWinFF or Ciano. Users who value an interface with a state-of-the-art look will probably prefer Curlew and FFQueue.

Keep in mind, however, that all the programs described in this article are resource hungry. Users with older systems that lack hyperthreading and multicore processing will have to wait for their output files. ■■■

Author

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

Info

[1] FFmpeg: *https://en.wikipedia.org/wiki/FFmpeg*

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

[3] Ciano: *https://robertsanseries.github.io/ciano/*

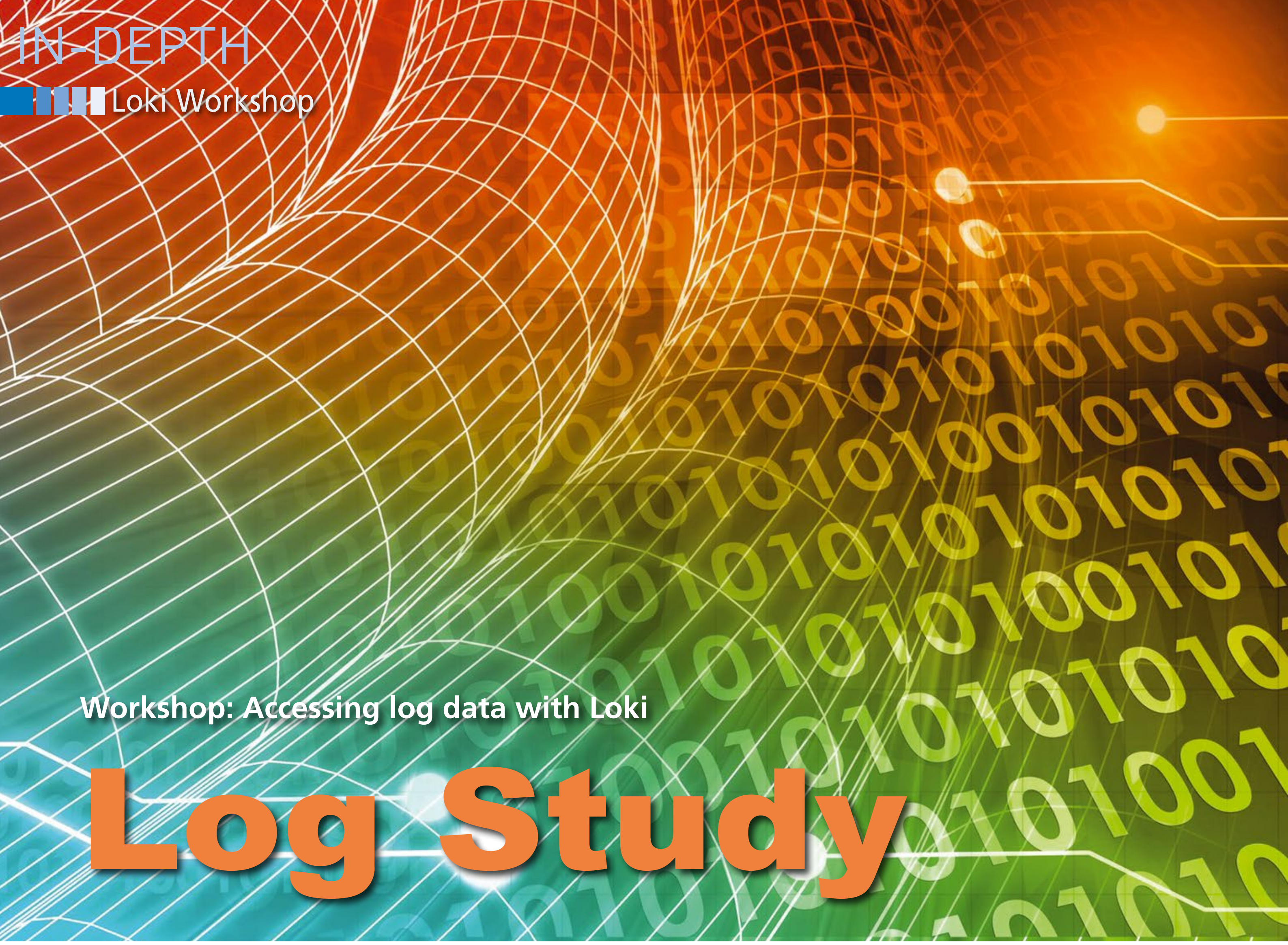
[4] Curlew: *https://curlew.sourceforge.io*

[5] FF Multi Converter: *https://sites.google.com/site/ffmulticonverter/*

[6] FFQueue: *https://ffqueue.bruchhaus.dk/Download.aspx*

[7] HandBrake: *https://handbrake.fr*

[8] QWinFF: *https://qwinff.github.io*



Workshop: Accessing log data with Loki

Log Study

Loki is a powerful, scalable, and easy-to-use solution for aggregating log data. *By Stefano Chittaro*

One day, during one of my company's cloud project meetings, a developer colleague said, "I need to find a way to quickly access logs for debugging and troubleshooting." I already had some experience with the Grafana-Prometheus, so I said I would help find a solution.

It turns out, the solution we settled on was Loki [1], from Grafana Labs. The Grafana Labs website describes Loki as "…a horizontally scalable, highly available, multi-tenant log aggregation system inspired by Prometheus." Loki is designed to aggregate logs efficiently, extracting metrics and alerts – all without requiring a massive indexing configuration. Once you have extracted the information you need, you can then use Grafana to visualize the data.

This workshop offers a quick look at how to access log data using Loki. In this scenario, I will push logs generated by an Apache web server hosting a sample Nextcloud deployment, then evaluate the data using Loki's own query language, LogQL.

In addition to a Loki server, I'll install the companion application Promtail [2], which Grafana Labs maintains as an agent to push data to Loki.

Log Factory

The first step is to start logging. If you already have a log-filled folder, just skip this step. In this case, I'll run a Nextcloud Docker container. I'll "mount" (bind) the Apache web server log folder inside the container to a local folder on the workstation. This step will allow Promtail, which will run locally, to access the files.

Of course you need to have the Docker engine installed. If you don't, see the box entitled "Get Docker Ready."

```
docker run --name nextcloud ⮒
-d -p 8080:80 -v /somelogsdir:⮒
/var/log/apache2 nextcloud
```

Once the container is running, you can sail a browser to:

```
http://localhost:8080
```

Get Docker Ready

This tutorial uses Docker as a way to spin up services quickly without installing unnecessary packages. In order to use Docker, you'll need the Docker engine, which you can install with a one-liner:

```
# curl -sSL https://get.docker.com |
  sudo bash
```

This command will fetch the latest official installation script. The script will detect which Linux distribution you're running, add the proper package manager repositories, and install.

Warning: Make sure you check the contents of a script every time you plan to pipe directly to `sudo bash`.

Lead Image © Kheng Ho Toh, 123RF.com

and perform some actions on the Nextcloud instance. It doesn't really matter what you do as long as it generates entries in the `access.log` and `error.log` Apache default log files.

Deploying

Download the binary release of the most recent version of both Loki and Promtail. Regarding Loki, fetch both the binary distribution and a sample config file:

```
wget ↩
https://raw.githubusercontent.com/↩
grafana/loki/v2.2.1/cmd/loki/↩
loki-local-config.yaml -O ↩
loki_config.yaml
```

```
wget https://github.com/grafana/↩
loki/releases/download/v2.2.1/↩
loki-linux-amd64.zip
unzip loki-linux-amd64.zip
```

The Loki sample config is already good enough for this workshop, so I'll execute it and keep it running:

```
./loki-linux-amd64 -config.file=↩
loki_config.yaml
```

If you see a bunch of `creating table` messages, that is a good sign – it means Loki is creating the structure to host the log entries.

The next step is to set up and run Promtail:

```
wget ↩
https://raw.githubusercontent.com/
grafana/loki/master/cmd/promtail/↩
promtail-local-config.yaml ↩
-O promtail_config.yaml

wget https://github.com/grafana/↩
loki/releases/download/v2.2.1/↩
promtail-linux-amd64.zip

unzip promtail-linux-amd64.zip
```

Before running Promtail, you'll need to

Listing 1: Promtail Configuration

```
01 server:
02   http_listen_port: 9080
03   grpc_listen_port: 0
04
05 positions:
06   filename: /tmp/positions.yaml
07
08 clients:
09    - url: "http://localhost:3100/
               loki/api/v1/push"

10 scrape_configs:
11 - job_name: apache
12   static_configs:
13   - targets:
14       - localhost
15     labels:
16       job: "apache"
17       instance: "localserver"
18       __path__: /somelogsdir/*.log
```

Structure

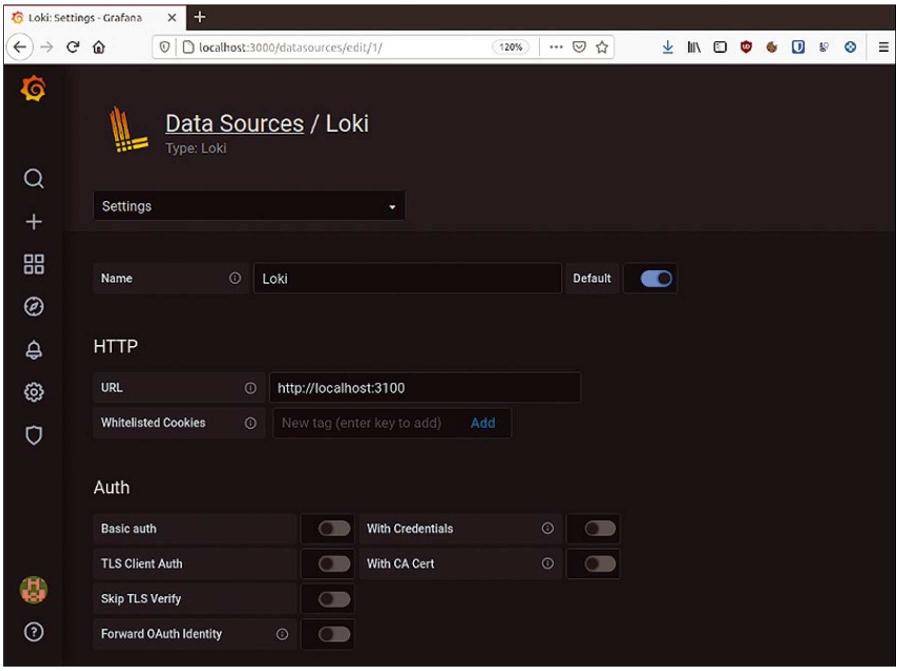
Once the logs are stored in Loki, they will be organized into streams. Each stream is identified by labels. Some labels are automatically generated (for example, "filename") and some are custom-made (See Listing 1, rows 16 and 17.)

In this case, I'll use {job="apache"} and {instance="localserver"}.

Loki will eventually store the log entries as pairs composed by a timestamp and the actual content.

Listing 2: Sample Log Query Result Object

```
{
  "status": "success",
  "data": {
    "resultType": "streams",
    "result": [
      {
        "stream": {
          "filename": "/somelogsdir/access.log",
          "instance": "localserver",
          "job": "apache"
        },
        "values": [
          [
            "1620748522681322318",
            "172.17.0.1 - - [11/May/2021:15:55:22 +0000] \"GET /csrftoken
             HTTP/1.1\" 200 928 \"-\"\"Mozilla/5.0 (X11; Ubuntu; Linux x86_64;
             rv:88.0) Gecko/20100101 Firefox/88.0\""
          ],
          [
            "1620748504911781382",
            "172.17.0.1 - - [11/May/2021:15:55:04 +0000] \"GET /csrftoken
             HTTP/1.1\" 200 929 \"-\"\"Mozilla/5.0 (X11; Ubuntu; Linux x86_64;
             rv:88.0) Gecko/20100101 Firefox/88.0\""
          ],
          [
            "1620748336477761747",
            "172.17.0.1 - - [11/May/2021:15:50:54 +0000] \"GET /cron.php
             HTTP/1.1\" 200 931 \"-\"\"Mozilla/5.0 (X11; Ubuntu; Linux x86_64;
             rv:88.0) Gecko/20100101 Firefox/88.0\""
          ]
        ]
      }
    ],
   "stats": {}
}
```



Name: Loki
Default: on
URL: http://localhost:3100
Whitelisted Cookies: New tag (enter key to add)
Basic auth: off
With Credentials: off
TLS Client Auth: off
With CA Cert: off
Skip TLS Verify: off
Forward OAuth Identity: off

Figure 1: Connecting the Grafana instance to Loki.

tweak the configuration to set up a Loki URL and log source folder (lines 9 and 18 in Listing 1).

Once you have successfully configured Promtail, run it with:

```
./promtail-linux-amd64 ⮐
-config.file=promtail_config.yaml
```

If everything is working as expected, you won't get any output yet from the logs being pushed to Loki.

Evaluation

Loki has no built-in UI, so the only way to query logs at this point is to make use of the excellent Loki RESTful API (see the box entitled "Structure").

The following query asks Loki to provide the most recent log entries, limiting the result to three entries:

```
curl -G -s  ⮐
"http://localhost:3100/loki/⮐
api/v1/query_range?limit=3" ⮐
--data-urlencode ⮐
'query={job="apache"}' | jq
```

The results of the query appear in Listing 2.

LogQL

Loki supports complex queries through (potentially) terabytes worth of logs.

You can compose your own queries using the LogQL query language, which is a modified version of the Prometheus language (PromQL). LogQL might look complicated at first, but you'll soon discover that it is basically a glorified grep.

LogQL queries consist of two parts:

- a log stream selector
- a log pipeline

Start by selecting one or more streams, and then apply a pipeline operator specifying the string you're looking for. For example, you might want to look for all accesses coming from a specific IP address:

```
{job="apache"} |= "172.17.0.1"
```

Or you might be interested in entries related to a specific page:

```
{job="apache"} |= "cron.php"
```

The following query defines the exact file you wish to search, loosely looking for Firefox accesses through regexp syntax:

```
{job="apache",filename=⮐
"/somelogsdir/access.log"} |~ ⮐
"Firefox.*"
```

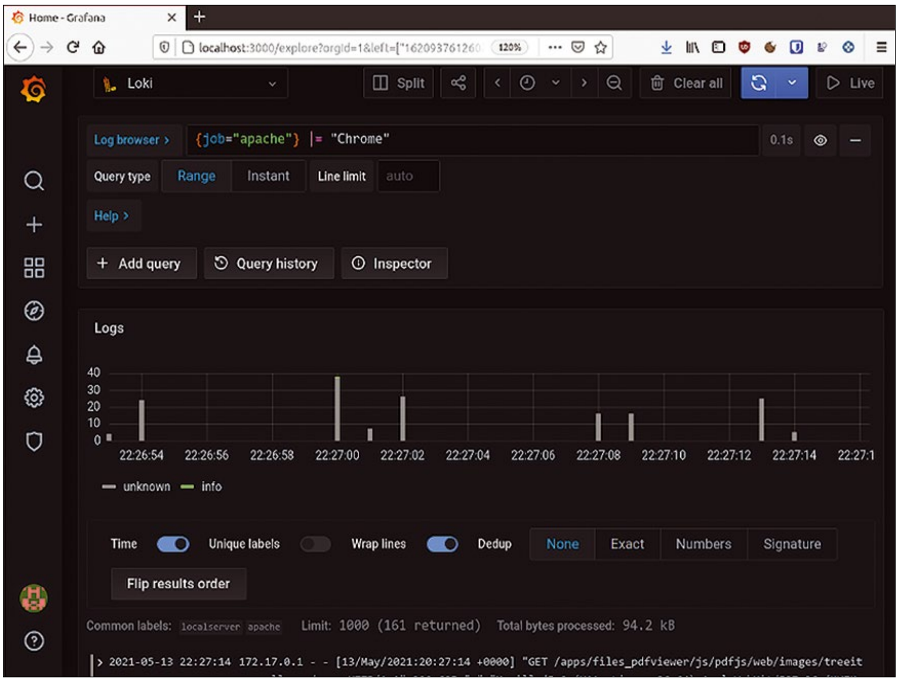
You can also elect *not* to filter specific strings. For instance, if you wish to exclude entries with an *http 200* status code from the result, use the ` character to delimit the search string:

```
{job="apache"} != `HTTP/1.1\" 200`
```

LogQL also provides a way to parse specific log formats (like JSON and logfmt) using a log parser. This option won't be useful in the case of Apache logs, but you'll find great documentation on the topic in the Loki wiki [3].

Grafana and Visualization

You can display the log data in a visually meaningful form using Grafana. To speed things up, I'll deploy Grafana through the official Docker image. I



```
{job="apache"} |= "Chrome"
```

Common labels: localserver apache Limit: 1000 (161 returned) Total bytes processed: 94.2 kB

```
> 2021-05-13 22:27:14 172.17.0.1 - - [13/May/2021:20:27:14 +0000] "GET /apps/files_pdfviewer/js/pdfjs/web/images/treeit
```

Figure 2: Comfortable querying and result sharing with Grafana.

will make sure to run it in *host* network mode so a connection to Loki is possible:

```
docker run -d ↩
--network host grafana/grafana
```

Once Grafana is up, use your browser to land on *http://localhost:3000*. Set up the default admin credentials, click on *Configuration* | *Data Sources*, and finally, select *Add Data Source*.

Select Loki as data source type and enter `http://localhost:3100` as the HTTP URL parameter (Figure 1) .

The *Save & Test* button will make sure settings are validated. You can finally move to the *Explore* tab, where you can freely query any data source (Figure 2). Once the query is entered and verified, share a short link with coworkers by clicking on the *Share* icon in the upper part of the page (again Figure 2).

Conclusions

Loki lets you set up a complete log aggregation infrastructure in a very short time span, without having to write a single line of code (see the box entitled "Loki vs Elasticsearch"). All components can run inside a Docker container or in a Kubernetes cluster when it's time to deploy Loki as a production application.

Loki also supports third-party storage for its logs collection (such as AWS S3 or Apache Cassandra). Next time you deploy a machine or a service, install a Promtail agent and give Loki a try. You'll be surprised by how quickly you can get productive. ■■■

Loki vs Elasticsearch

In the log aggregation ecosystem, Elasticsearch (and, in general, the ELK stack) is certainly a popular choice, but in my opinion, Loki might be better for certain use cases. I give the advantage to Loki for:

* Scalability: Although Elasticsearch indexes all elements of log entries beforehand, Loki specializes in brute force text searches. Loki data is stored unstructured, meaning that Loki can handle a larger amount of data compared to Elasticsearch.

* Metrics format: Loki stores logs with the same structure logic as Prometheus TSDB (streams). This approach means that an application stack (Grafana, Prometheus, and Loki) can pinpoint an application issue starting from a metric or the other way around.

The Loki project has a comparison page for your consideration [4]. Always choose the best tool for your use case.

Info

[1] Loki: *https://grafana.com/oss/loki/*

[2] Promtail: *https://grafana.com/docs/loki/latest/clients/promtail/*

[3] LogQL Reference: *https://grafana.com/docs/loki/latest/logql/*

[4] Loki vs Elasticsearch: *https://grafana.com/docs/loki/latest/overview/comparisons/*

Author

Stefano Chittaro manages multicloud deployments with a special focus on automation and observability.

GETTING STARTED WITH LINUX
LibreOffice
Become a LibreOffice Expert!
301 BEST BASH COMMANDS
LINUX SHELL HANDBOOK
SUPERCHARGE YOUR LINUX SKILLS
101 COOL LINUX HACKS

The sys admin's daily grind: exa

Show Me!

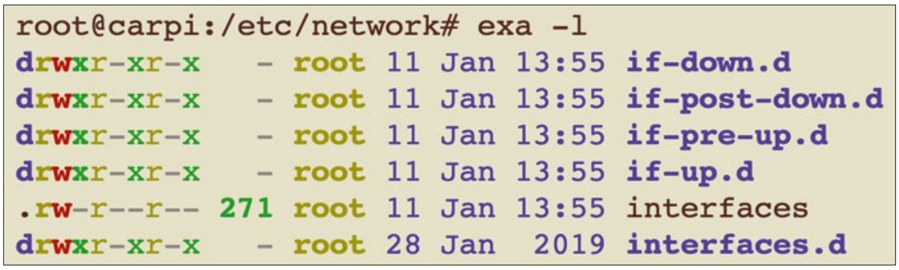
There is nothing that admins hate more than unnecessary typing at the console. That's why Charly's clever alternative exa replaces the classic, but ancient, ls. *By Charly Kühnast*

There are very few commands that you type more often than `ls` when working in the shell. As experienced Linux users will know, the shorter a command is, the more important it is. Even the often expressed suspicion that the shortest commands have the longest man pages is something that `ls` cannot really refute. You would think that there is little to improve with `ls`, but the `exa` developers beg to differ.

Instead of a one-to-one replacement, `exa` [1] seeks to be the better `ls`. It gets by with far fewer parameters and is correspondingly less powerful. But if you use only the most common options – for me, that would be `-l`, `-a`, `-t`, `-h`, and `--sort=size` – you won't miss anything. That's because `exa` comes with sensible defaults that make some parameters superfluous. For example, `exa` always displays file sizes in a notation that is easily readable for humans, such as `2.9M` instead of `2893342`, which you first need to enable by adding `-h` in `ls`.

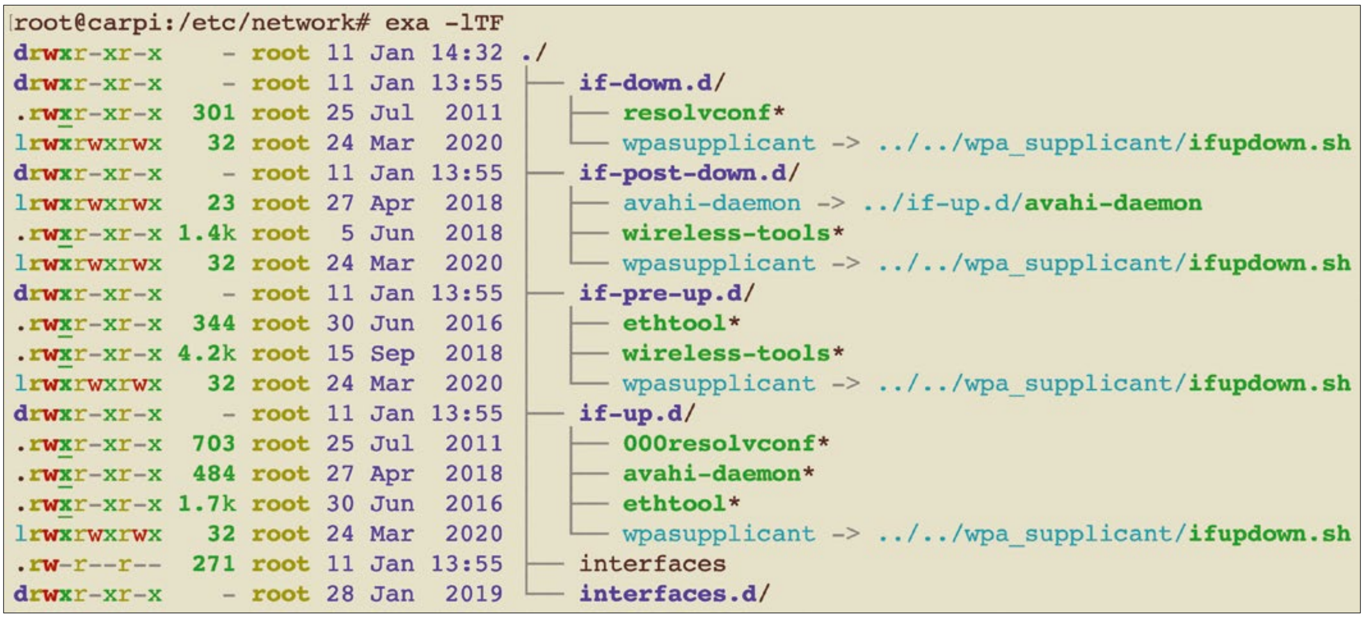
The output from `exa -l`, the counterpart of `ls -l`, looks like Figure 1. This is still very reminiscent of the original, except for the lush coloring. Using the additional `-F` parameter (for "file type"), I can show additional characters that indicate the file type. For example, an asterisk is appended to executable files, a slash to directories, and an at sign to symbolic links. The `sort` parameter, which is very important for me personally, also works as expected. The command `--sort=size` sorts by file size, while `-r` inverts the sort order.

Sorting by time also works as expected – you can even choose whether you want `exa` to sort the list by `--time=created`, `--time=modified`, or `--time=accessed`. The tree view (`-T`) shows you the directory tree including the subdirectories (Figure 2), while the `--level=` parameter tells `exa` how deep you want it to dig down. ■■■



```
root@carpi:/etc/network# exa -l
drwxr-xr-x   - root 11 Jan 13:55 if-down.d
drwxr-xr-x   - root 11 Jan 13:55 if-post-down.d
drwxr-xr-x   - root 11 Jan 13:55 if-pre-up.d
drwxr-xr-x   - root 11 Jan 13:55 if-up.d
.rw-r--r-- 271 root 11 Jan 13:55 interfaces
drwxr-xr-x   - root 28 Jan  2019 interfaces.d
```

Figure 1: The output from `exa -l`, the counterpart to `ls -l`.



```
root@carpi:/etc/network# exa -lTF
drwxr-xr-x    - root 11 Jan 14:32 ./
drwxr-xr-x    - root 11 Jan 13:55 ├── if-down.d/
.rwxr-xr-x  301 root 25 Jul  2011 │  ├── resolvconf*
lrwxrwxrwx   32 root 24 Mar  2020 │  └── wpasupplicant -> ../../wpa_supplicant/ifupdown.sh
drwxr-xr-x    - root 11 Jan 13:55 ├── if-post-down.d/
lrwxrwxrwx   23 root 27 Apr  2018 │  ├── avahi-daemon -> ../if-up.d/avahi-daemon
.rwxr-xr-x 1.4k root  5 Jun  2018 │  ├── wireless-tools*
lrwxrwxrwx   32 root 24 Mar  2020 │  └── wpasupplicant -> ../../wpa_supplicant/ifupdown.sh
drwxr-xr-x    - root 11 Jan 13:55 ├── if-pre-up.d/
.rwxr-xr-x  344 root 30 Jun  2016 │  ├── ethtool*
.rwxr-xr-x 4.2k root 15 Sep  2018 │  ├── wireless-tools*
lrwxrwxrwx   32 root 24 Mar  2020 │  └── wpasupplicant -> ../../wpa_supplicant/ifupdown.sh
drwxr-xr-x    - root 11 Jan 13:55 ├── if-up.d/
.rwxr-xr-x  703 root 25 Jul  2011 │  ├── 000resolvconf*
.rwxr-xr-x  484 root 27 Apr  2018 │  ├── avahi-daemon*
.rwxr-xr-x 1.7k root 30 Jun  2016 │  ├── ethtool*
lrwxrwxrwx   32 root 24 Mar  2020 │  └── wpasupplicant -> ../../wpa_supplicant/ifupdown.sh
.rw-r--r--  271 root 11 Jan 13:55 ├── interfaces
drwxr-xr-x    - root 28 Jan  2019 └── interfaces.d/
```

Figure 2: The tree view in `exa` visualizes the directory structures.

Info

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

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.

FREE DVD!
LibreOffice Full Version

Become a LibreOffice Expert!

2020 Edition

LibreOffice

Dive deep into the world's greatest free office suite

Write Your Own LO Macros
Save time and automate common tasks

Digital Signatures
Lock down your private documents

Edit and Save MS Office Files

Create Professional:
- Text Documents
- Spreadsheets
- Presentations
- Databases

Replace MS Office and Google Docs!

LibreOffice

Includes full versions for Windows, macOS, and Linux

WWW.LINUX-MAGAZINE.COM

LibreOffice 7.0.2
Installation files for Windows, macOS, and Linux
Lots of extras inside! ■ Handbooks ■ Extensions ■ Videos

LIBREOFFICE 2020
LINUX MAGAZINE

Fast Debian downloads

Scripted

By combining a few Debian scripts, you can increase your download speed by more than 50 percent. Bruce shows you how. *By Bruce Byfield*

Over the years, a rich ecosystem of applications has grown around Debian package management. In fact, the available choices are so numerous that some are easy to overlook, especially since some scripts are not in the official repositories of any distribution. Finding compatible scripts can require a lot of searching, often resulting in frustration.

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/*.

Setting out to maximize download speed, I found several layers of scripts or tools: netselect [1] to find the quickest mirrors for my location, aria2 [2] or Axel [3] to serve as download managers, and apt-fast [4] to make the other scripts' modifications easy to use by placing them in a single command. The procedure can take some time, but, considering that it can increase download speeds by more than 50 percent, the effort seems worthwhile.

Step One: Determine the Fastest Mirrors

In earlier versions of Ubuntu, you could have the fastest mirror selected for you by choosing *Settings* | *Repositories* | *Download* | *Other* | *Best Server*. However, this feature has been dropped in the past few years.

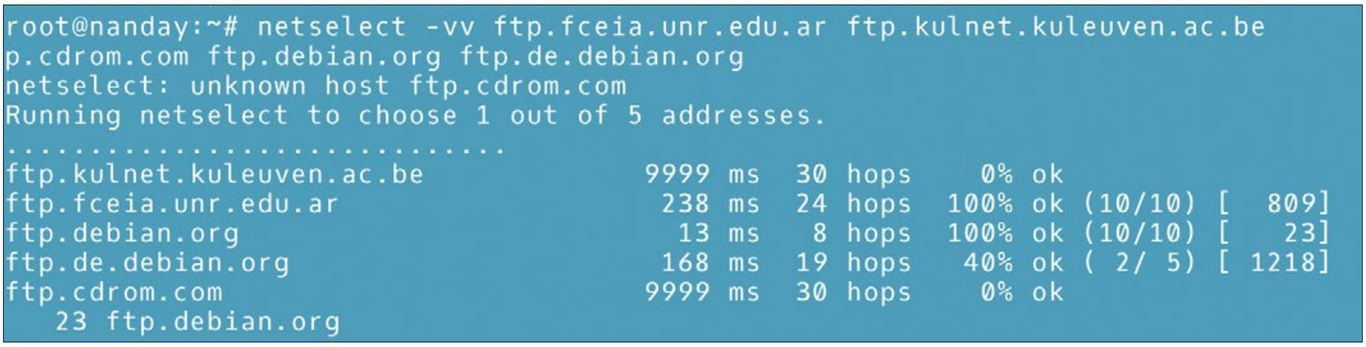
Netselect offers a better solution for finding the fastest repositories. Not only does it give you multiple mirrors, which you will want when using download managers, netselect also allows you to consider the factors involved, such as how close a repository is to you, its bandwidth, and its traffic at various times of day. Perhaps no one repository will always be the fastest, but you can check at different times and get an average reading of its speed.

When you research how to find the fastest server, you will find frequent references to netselect-apt. This script is still available and has the advantage of selecting the fastest mirrors for you. However, installing it on a modern Debian-like system results in dependency problems you may not want to

Listing 1: Your Top 10 Fastest Links

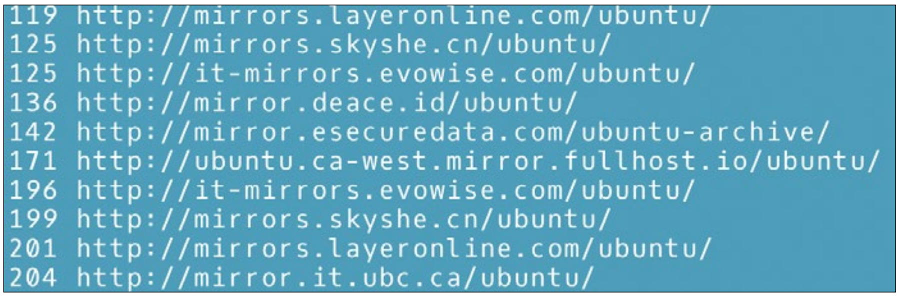
```
sudo netselect -v -s10 -t20 `wget -q -O- https://launchpad.net/ubuntu/+archivemirrors | grep -P -B8 "statusUP|statusSIX" |
grep -o -P "(f|ht)tp://[^\"]*"`
```

Lead Image © Kjetil Kolbjornrud, 123RF.com



```
root@nanday:~# netselect -vv ftp.fceia.unr.edu.ar ftp.kulnet.kuleuven.ac.be
p.cdrom.com ftp.debian.org ftp.de.debian.org
netselect: unknown host ftp.cdrom.com
Running netselect to choose 1 out of 5 addresses.
..............................
ftp.kulnet.kuleuven.ac.be              9999 ms  30 hops    0% ok
ftp.fceia.unr.edu.ar                    238 ms  24 hops  100% ok (10/10) [  809]
ftp.debian.org                           13 ms   8 hops  100% ok (10/10) [   23]
ftp.de.debian.org                       168 ms  19 hops   40% ok ( 2/ 5) [ 1218]
ftp.cdrom.com                          9999 ms  30 hops    0% ok
   23 ftp.debian.org
```

Figure 1: Use netselect to find the fastest mirrors from an input list.



```
119 http://mirrors.layeronline.com/ubuntu/
125 http://mirrors.skyshe.cn/ubuntu/
125 http://it-mirrors.evowise.com/ubuntu/
136 http://mirror.deace.id/ubuntu/
142 http://mirror.esecuredata.com/ubuntu-archive/
171 http://ubuntu.ca-west.mirror.fullhost.io/ubuntu/
196 http://it-mirrors.evowise.com/ubuntu/
199 http://mirrors.skyshe.cn/ubuntu/
201 http://mirrors.layeronline.com/ubuntu/
204 http://mirror.it.ubc.ca/ubuntu/
```

Figure 2: A netselect list of the top 10 fastest Ubuntu links.

Table 1: Fields in an /etc/axelrc File

Field	Description
reconnect_delay	Sets the number of seconds Axel waits before trying to reconnect.
max_speed	Value is given in bytes per second and dependent on the physical bandwidth.
num_connections	Sets the maximum number of connections that Axel will attempt to use. Some servers may not permit multiple connections.
http_proxy	Sets a proxy server if no system-wide environmental variable is set.
no_proxy	Provides an optional comma-separated list of local domains that Axel should not try to reach through a proxy.
buffer_size	Represents the maximum bytes to read from all connections at the same time.
verbose	Set to 0 to disable screen messages and 1 to see messages.
interfaces	Lists network interfaces with Internet access. If not specified, the first interface found will be used.

deal with. Instead, you should use the newer netselect script. If you are interested in particular mirrors, you can check their address in Debian's list of mirrors [5] or in your distribution's list and enter them in a space-separated list (Figure 1). The output will show the speed of each mirror address in parentheses at the end of its line, and the fastest in the final line, with the lowest score being the fastest. You can also use options to limit the listings and the number of fast mirrors returned. Alternatively, you can get a list of the 10 fastest links (Figure 2) for your circumstances with the code from Listing 1.

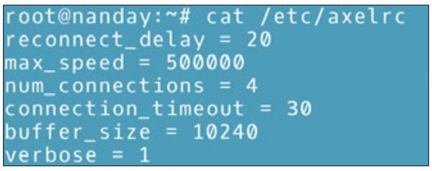
Be warned though, that almost 600 mirrors will be checked using the Debian list alone, and the results will take a few minutes to process. Either way, you will want at least five mirrors for the next steps in this procedure. An alternative method for finding the fastest mirrors is apt-smart [6], which can also manage package sources.

Step 2: Set Up a Download Manager

Adding the fastest mirrors to /etc/apt/sources.list accelerates downloads without any other modification. However, you can increase the download speed even more by adding a download manager such as Axel or aria2. Both are replacements for Wget and available for non-Debian systems, but what is relevant here is that both will attempt to use multiple mirrors to accelerate downloads.

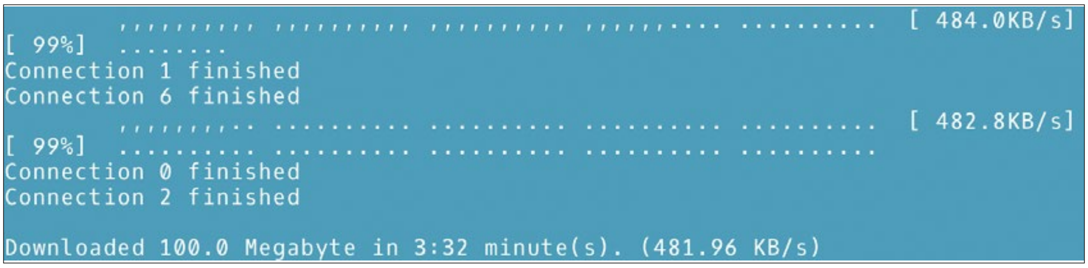
Axel can be configured with options, or, more conveniently, with the fields in /etc/axelrc (Table 1).

Figure 3 shows a sample minimal axelrc file. You may want to add additional fields, depending on your hardware. Running Axel verbosely, the output shows the percentage of the download completed and ends with the connections closing and a download summary (Figure 4).



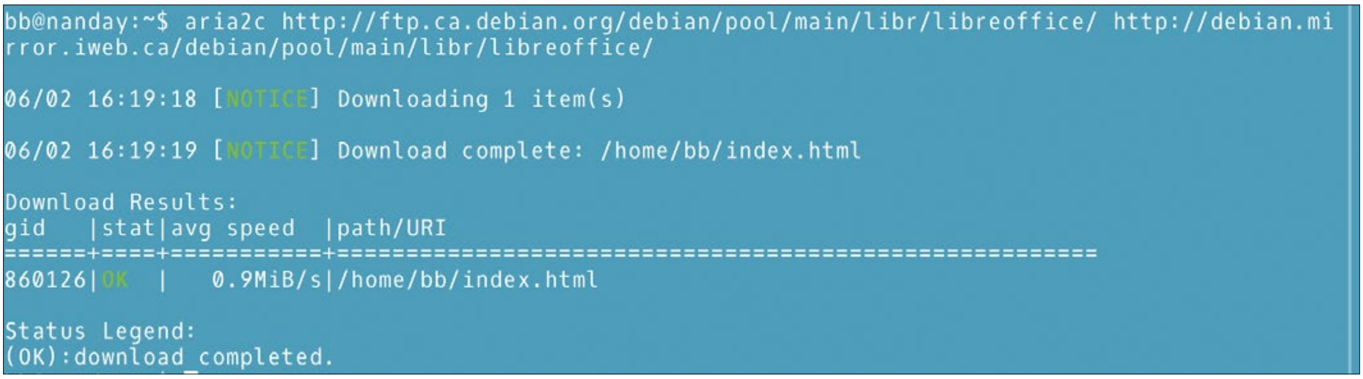
```
root@nanday:~# cat /etc/axelrc
reconnect_delay = 20
max_speed = 500000
num_connections = 4
connection_timeout = 30
buffer_size = 10240
verbose = 1
```

Figure 3: Configure Axel by creating the /etc/axelrc file.



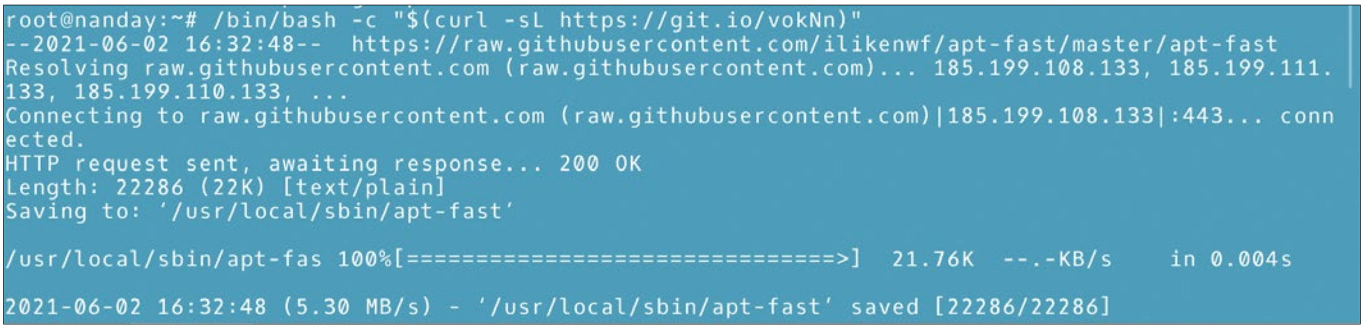
```
     ,,,,,,,,,, ,,,,,,,,,, ,,,,,,,,,, ,,,,,,,.... ..........  [ 484.0KB/s]
[ 99%]  ........
Connection 1 finished
Connection 6 finished
     ,,,,,,,,.. .......... .......... .......... ..........  [ 482.8KB/s]
[ 99%]  .......... .......... .......... .......... ..........
Connection 0 finished
Connection 2 finished

Downloaded 100.0 Megabyte in 3:32 minute(s). (481.96 KB/s)
```

Figure 4: Axel ends with a summary of its operations.



```
bb@nanday:~$ aria2c http://ftp.ca.debian.org/debian/pool/main/libr/libreoffice/ http://debian.mi
rror.iweb.ca/debian/pool/main/libr/libreoffice/

06/02 16:19:18 [NOTICE] Downloading 1 item(s)

06/02 16:19:19 [NOTICE] Download complete: /home/bb/index.html

Download Results:
gid   |stat|avg speed  |path/URI
======+====+===========+=======================================================
860126|OK  |   0.9MiB/s|/home/bb/index.html

Status Legend:
(OK):download completed.
```

Figure 5: In aria2, multiple mirrors can be added to the command-line options.



```
root@nanday:~# /bin/bash -c "$(curl -sL https://git.io/vokNn)"
--2021-06-02 16:32:48--  https://raw.githubusercontent.com/ilikenwf/apt-fast/master/apt-fast
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.111.
133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... conn
ected.
HTTP request sent, awaiting response... 200 OK
Length: 22286 (22K) [text/plain]
Saving to: '/usr/local/sbin/apt-fast'

/usr/local/sbin/apt-fas 100%[===============================>]  21.76K  --.-KB/s    in 0.004s

2021-06-02 16:32:48 (5.30 MB/s) - '/usr/local/sbin/apt-fast' saved [22286/22286]
```

Figure 6: Apt-fast has a download script for installation.

As an alternative to Axel, you may want to use aria2. In aria2, multiple mirrors are added at the command line or, as shown in Figure 5, in a file specified using the option:

```
--input-file=FILE (-i=FILE)
```

Naturally, you will want to use the fastest mirrors as determined by netselect (see above). However, unlike Axel, aria2 does not show how each mirror was used. See Table 2 for other useful aria2 options.

Step 3: Set Up apt-fast

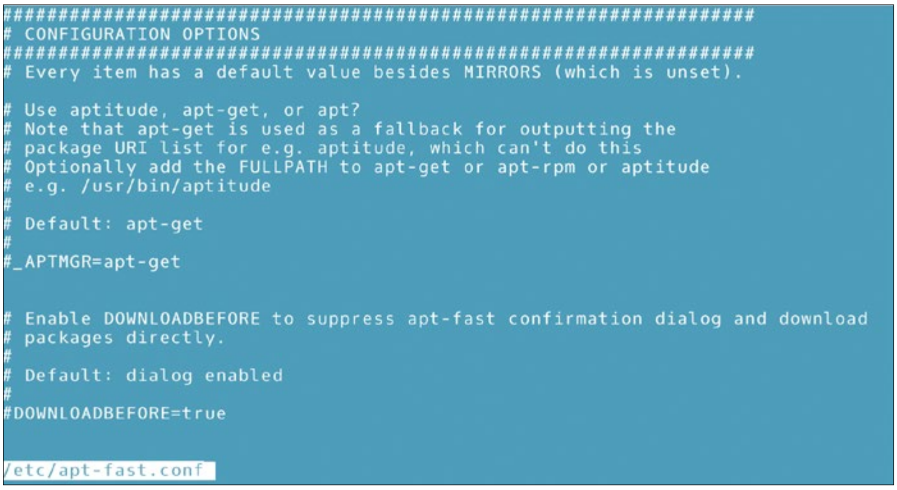
Apt-fast offers the ultimate quick download (Figure 6). This script is not in official Debian or Ubuntu repositories, but you can install it using the script:

```
/bin/bash -c "$(➘
  curl -sL https://git.io/vokNn)"
```

Once apt-fast is installed, edit `/etc/apt-fast.conf` in a text editor (Figure 7). Options include whether to use `apt-get/apt` or the Debian/Ubuntu mirrors and to list your fastest mirrors in a space or comma-separated list. For example:

```
MIRRORS=( ➘
  'http://ftp.debian.org/debian, ➘
  http://ftp2.de.debian.org/debian, ➘
  http://ftp.de.debian.org/debian, ➘
  ftp://ftp.uni-kl.de/debian' )
```

The same mirrors must also be in your `/etc/apt/sources.list`. You can also scroll down the file and check whether



```
#######################################################################
# CONFIGURATION OPTIONS
#######################################################################
# Every item has a default value besides MIRRORS (which is unset).

# Use aptitude, apt-get, or apt?
# Note that apt-get is used as a fallback for outputting the
# package URI list for e.g. aptitude, which can't do this
# Optionally add the FULLPATH to apt-get or apt-rpm or aptitude
# e.g. /usr/bin/aptitude
#
# Default: apt-get
#
#_APTMGR=apt-get


# Enable DOWNLOADBEFORE to suppress apt-fast confirmation dialog and download
# packages directly.
#
# Default: dialog enabled
#
#DOWNLOADBEFORE=true


/etc/apt-fast.conf
```

Figure 7: The most efficient way to use apt-fast is with its configuration file.

Table 2: aria2 Options

Option	Description
`--max-overall-upload-limit=SPEED`	Sets the maximum overall speed in bytes, kilobytes, or megabytes.
`--connect-timeout=SECONDS`	Sets the connect timeout in seconds. Default is `60`; `0` means no timeout
`--max-file-not-found=NUMBER`	Sets the maximum number of files not found before download fails.
`--max-tries=NUMBER`	Sets number of tries before download fails. `0` means unlimited tries.
`--retry-wait=NUMBER`	Sets the seconds to wait between retries. The default is 5.

other fields, such as the number of connections per command, should be edited. Check, too, that Axel or aria2 is installed.

To use apt-fast, the basic command structure is:

```
apt-fast install PACKAGE
```

The command starts by listing all the files that will be installed (Figure 8). When all packages are downloaded, download statistics are displayed (Figure 9). The rest of the output then follows standard `apt-get` output as packages are unpacked and installed. Note that the script is only used for package downloads, and that other `apt` functions are performed normally.

Last Considerations

Whether any or all of these scripts are worth setting up depends on what you download. If you regularly download ISO images, these scripts are worth considering. However, if your downloads are smaller, the scripts may not be worth your time if you have a modern computer – unless, of course, you are the kind of user who likes to squeeze every bit of speed from your hardware operations. Be aware, too, that how much faster you can make your downloads depends on your hardware and circumstances. In the best cases, you might double or triple your download speed. If you do try these scripts, make sure you back up `/etc/apt/sources.list`, just in case. ■■■

Info

[1] netselect: *https://packages.debian.org/sid/net/netselect*

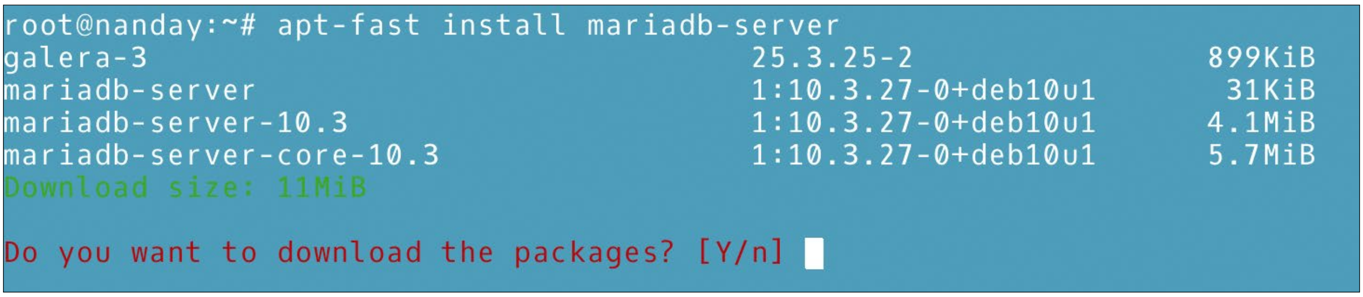
[2] aria2: *https://aria2.github.io/*

[3] Axel: *https://linux.die.net/man/1/axel*

[4] apt-fast: *https://github.com/ilikenwf/apt-fast*

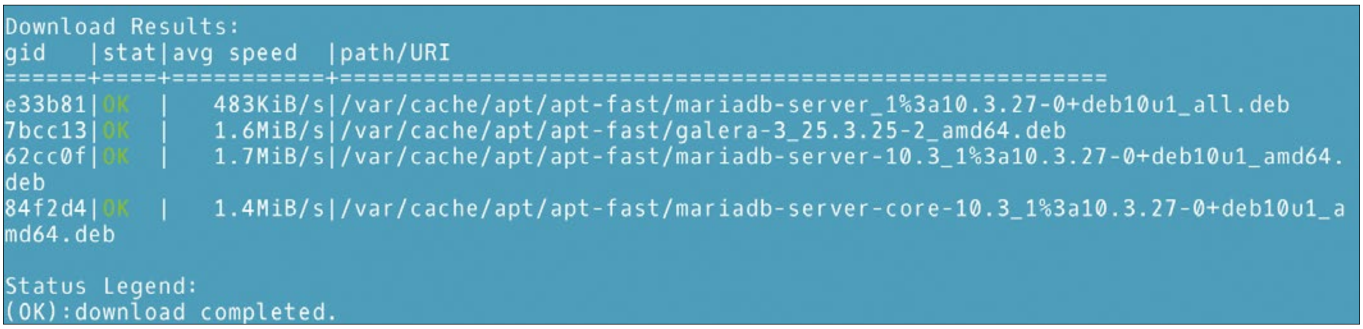
[5] debian mirror list: *https://www.debian.org/mirror/list*

[6] apt-smart: *https://github.com/martin68/apt-smart*



```
root@nanday:~# apt-fast install mariadb-server
galera-3                                25.3.25-2                   899KiB
mariadb-server                          1:10.3.27-0+deb10u1          31KiB
mariadb-server-10.3                     1:10.3.27-0+deb10u1         4.1MiB
mariadb-server-core-10.3                1:10.3.27-0+deb10u1         5.7MiB
Download size: 11MiB

Do you want to download the packages? [Y/n]
```

Figure 8: The start of an apt-fast download.



```
Download Results:
gid   |stat|avg speed  |path/URI
======+====+===========+=======================================================
e33b81|OK  |   483KiB/s|/var/cache/apt/apt-fast/mariadb-server_1%3a10.3.27-0+deb10u1_all.deb
7bcc13|OK  |   1.6MiB/s|/var/cache/apt/apt-fast/galera-3_25.3.25-2_amd64.deb
62cc0f|OK  |   1.7MiB/s|/var/cache/apt/apt-fast/mariadb-server-10.3_1%3a10.3.27-0+deb10u1_amd64.
deb
84f2d4|OK  |   1.4MiB/s|/var/cache/apt/apt-fast/mariadb-server-core-10.3_1%3a10.3.27-0+deb10u1_a
md64.deb

Status Legend:
(OK):download completed.
```

Figure 9: When packages are downloaded, apt-fast summarizes results.

An interview with IRCNow's Aaron Lin

The Users' Network

IRCNow empowers users and lets them break free from social media giants. Project leader Aaron Lin shares his vision. *By Rubén Llorente*

IRCNow, a communication platform built on free and open source software and open standards, transforms Internet Relay Chat (IRC) into a user-controlled social network. IRCNow strives to give users as much control of the software stack as possible, with the goal of being vendor independent. With 13 active servers at the time of writing, IRCNow functions as a federation of self-governing servers.

Aaron Lin (Figure 1), who has directed IRCNow [1] since its creation in 2019, recruits and trains volunteers and teaches them how to run the network. In this interview, he provides the back story on IRCNow and reveals what the future holds for the network.

Linux Magazine: What is IRCNow, and why is it important?

Aaron Lin: IRCNow aims to be a social network that the users control. What IRCNow does differently from most social networks is we educate our users and get them involved in self-governance. We have courses to teach our users system administration [Figure 2], and we encourage them to run their own servers. We also plan to write courses to teach them how to code. We believe that the only way to build a free community is to make sure users are educated and have control over their code.

We chose the IRC protocol because it is the most accessible to average users. We intend to make IRC suitable for users of any age, skill level, and creed.

In addition, we are aiming to be fully independent from corporate vendors. More people are beginning to realize that big technological corporations cannot be trusted to respect their users, and we want to be a home for those people.

LM: So IRCNow is all about letting users control the software they communicate with. Don't projects such as Mastodon or Matrix pursue the same goal? Why start yet another project?

AL: Our sense is that Mastodon and Matrix are not as focused on educating their users to be sys admins or coders. The majority of instances appear to be one-click installs. This is convenient, but users don't get to see how to do basic system administration, as well as how to write their own code. This leaves them at the mercy of upstream developers. Mastodon and Matrix give the user more control over the platform than Twitter, but you are still left with little power to influence development.

There have even been discussions at Mastodon's GitHub site about hard coding bans on instances that they disagree with [2]. There have also been Mastodon clients that hard code such bans [3]. If users lack the ability to code for themselves, they have to accept whatever limitations the developers place on it [the site], which means they have lost control of the platform.

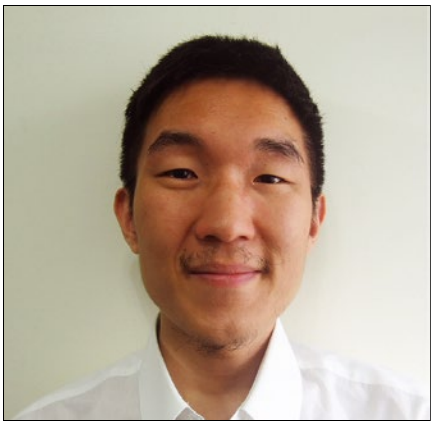


Figure 1: Aaron Lin has managed the IRCNow project since 2019.

Lead Image © Giordano Aita, 123RF.com

Figure 2: IRCNow actively recruits novices and turns them into system administrators.

The technical barriers of Matrix and Mastodon make it difficult for a user community of amateurs to have this level of control over the code. IRCNow focuses on simple, more mature protocols like IRC, because the protocol is really easy to learn to code for, and these are very accessible protocols with mature, cross-platform clients.

LM: When you say Matrix and Mastodon present technical barriers that make it difficult for users to control the code, which sort of barriers are you talking about?

AL: Both Matrix and Mastodon are designed for the web. They assume the use of HTTP, and much of their ecosystem is based around web apps. Naturally, the majority of their users are accessing these platforms using web apps such as Element. Web apps often require more sophisticated coding skills. Another problem is that, with a web app, the browser becomes an important part of the solution, and web apps frequently break and malfunction when used with free software browsers that are controlled by the community.

If you take a look at the mainstream browsers – Firefox, Chrome, Edge, Safari – all of them have serious censorship and privacy issues. Many of them have non-free components. Average users have very little real control over the development of the software. What is worse, the controlling organizations often have agendas that run contrary to the interests of their user community.

LM: You place a lot of emphasis on training users so they are able to develop and run their own software in order to make them less reliant on software providers. How are you achieving this?

AL: We're working on a few ways to improve tech literacy. First, we're offering free shell accounts and tutorials to help users learn about Unix. The tutorials will try to teach the basics of how to use a text editor, how Unix pipes work, how to read manual pages, and so forth [see the "IRCNow Training Camp" box].

Once users pass that basic training, we offer them free VPSs if they can help us run our network. We train them in the applications bundled in the OpenBSD base. We walk them through setting up their own web server, requesting a TLS cert, configuring a mail server, and [configuring a] name server. We also help them install and configure an IRC server and bouncer. Because it is OpenBSD, trainees are forced to read the documentation. Our teammates are there to help with tech support. If they do well, we try to upgrade them to help manage dedicated servers.

The next stage we plan to work on is to teach our users how to code. We have not yet begun this, but I plan to cover C, Perl, and Korn shell scripts using real practical code that can be run. I chose these three languages because they are what OpenBSD itself is written in. If the need should ever arise, our users would have the ability to fork the operating system.

LM: The goal then is to have the users control as much of their social media's software stack as possible. It is a very ambitious project! Where are the resources coming from?

AL: About 15 percent of the costs have come from individual donors and 85 percent from my personal funds. Currently, we are spending roughly $5,000 per year in hosting fees. We're grateful for all the donors who have helped us with donations in hardware, bandwidth, and time. If your readers would like to support the project, we'd greatly appreciate [their support].

Our goal is to eventually migrate over to a combination of user subscriptions and third-party transaction fees.

LM: A quick glance at your wiki suggests IRCNow is being built using OpenBSD [4] and its userspace tools. Why use OpenBSD instead of another popular free and open source operating system?

AL: I chose OpenBSD because of its focus on simple configuration, correct code, good documentation, and its commitment to software freedom. Contrary to popular belief, it's actually a great operating system for novices to learn about system administration and coding.

There's this misconception that Linux is "easier." This depends what you mean by easier. It's true that a distro like Ubuntu would allow you to quickly install black-box software without the need to read documentation. But later, when you need to troubleshoot, customize, or fork that software, you discover it is breathtakingly complex.

Our users need real open source: They need to open the source and start coding! OpenBSD gives us a system where our users could someday have full control over [the system], with the ability to customize or fork where needed.

LM: What is the biggest challenge IRCNow has faced?

AL: The greatest challenge is educating users. Having an educated user base is not something you can purchase or automate and deploy. Our teammates have spent thousands of hours of labor trying to teach users how to use shell accounts, how to read manual pages, and how to troubleshoot their servers. We plan to spend thousands more teaching them how to code. Courseware and labs are essential and have their place, but there is no substitute for the human touch.

It's a common mistake to believe the value of a network is in the technology, in the software or the hardware. All infrastructure is easily replaceable. The greatest asset of any network is its users and their culture. Most of our efforts are spent educating users about the need for digital freedom, about the importance of hosting our own infrastructure. Our network welcomes everyone, but we do not want users to remain tech-illiterate. If we don't educate our users, we will end up recreating the same degrading culture that exists on most mainstream social networks. We need to empower our users so they can be involved in running our network.

Whether IRCNow succeeds or fails will depend on how well we can preserve this culture of education and commitment to user freedom. ■■■

IRCNow Training Camp

IRCNow makes an effort to bring new users to the network, including people with absolutely no experience with Unix-like systems or system administration. The IRCNow developers are creating a training course that is still a work in progress, although early indications are that it will be a very good introduction to the IRC environment.

With six phases currently implemented, the first three give a basic understanding of common command-line tools, what a shell is, and how to navigate the filesystem tree. The fourth explains how to use pipes and also introduces man pages.

Especially outstanding, the fifth phase offers a funny (yet instructive) vi tutorial. Given IRCNow's commitment to keep things simple, it uses vi instead of Vim, because OpenBSD's base includes vi.

With an eminently practical bent, the sixth phase teaches about terminal multiplexing with tmux, using Lynx, and practicing with Irssi.

Trainees who make it this far are usually then granted a free virtual private server (VPS) and mentored into deploying common services, such as websites and email systems.

Info

[1] IRCNow: *https://www.ircnow.org/*

[2] Discussion about banning Gab from Mastodon: *https://github.com/tootsuite/mastodon/issues/11129*

[3] Tusky hard-codes ban against Gab: *https://github.com/tuskyapp/Tusky/pull/1303*

[4] OpenBSD: *https://www.openbsd.org*

Author

Rubén Llorente is a mechanical engineer who ensures that the IT security measures for a small clinic are both legally compliant and safe. In addition, he is an OpenBSD enthusiast and a weapons collector.

ADMIN Network & Security
AD Protected Users
Web Servers
LEAN WEB Servers
Go lighter and get faster
Lock Down Microsoft IIS
Secure Containers
Secure Containers with a hypervisor DMZ
.NET Core
Securing TLS
Securing TLS Put an end to fake certificates
Layer 3 SDN
Custom MIBs
Custom MIBs Get smarter with system monitoring
LINSTOR
Graph Databases
Graph Databases
Ransomware Strategies
Automation
Automation Get ready for DevOps with Ansible, CFEngine, Chef, Puppet, and SaltStack
FREE DVD

The new Python match

DIVERSIONS

Exploring the new Python match statement, Python's implementation of switch/case.

By Scott Sumner

If you've decided to learn Python [1] and have any experience with other programming languages, you'll quickly notice that the ubiquitous `switch` statement is nowhere to be found. Soon, though, that will no longer be the case (sort of). Python 3.10 is slated to be released in October 2021 and includes the new `match` command [2] [3].

Switcheroo

The function of `switch` is akin to trying to find a particular office in an office building. As you walk down the hallway, you look at each door to see if it displays the number or name in which you are interested. When you find it, you stop searching and go inside. In C (and indeed many other languages) `switch` allows you to compare a value against a set of others (Listing 1).

The `switch` starts the compare operation, with the value you want to check passed in. The sets of curly braces then contain your `case` statements.

Each `case` is a value against which to check, so in this example, `case 1:` says "if the value is 1, run this block of code." The program steps through each `case` statement until it finds a match and then runs the code inside that block. Here, I've only put in a single print statement, but the code block can contain as many statements as you want. When the program encounters a `break`, it exits the `switch` statement, and the program moves on to whatever follows the `switch` block.

You'll also notice `default:` near the bottom of the example, which runs if none of the other `case` statements match. Although not required, if `default` is not there and no `case` matches the `switch` value, your program will move on to whatever follows the `switch` statement without running anything for that block.

Python match

Until now, Python didn't really have a counterpart to the `switch` statement, so you had to employ a string of `if` and `elif` statements:

```
x = 3
if x == 1:
  print ( "1" )
elif x == 2:
  print ( "2" )
elif x == 3:
  print ( "3" )
else:
  print ( "Not 1, 2, or 3" )
```

The same code with `match` is:

```
x = 3
match x:
  case ( 1 ):
    print ( "1" )
  case ( 2 ):
    print ( "2" )
  case ( 3 ):
```

Listing 1: switch and case in C

```
x=3
switch ( x )
{
  case 1:
  {
    printf ( "x is 1" );
  }
  break;
  case 2:
  {
    printf ( "x is 2" );
  }
  break;
  case 3: printf ( "x is 3" );
  break;
  default: printf
  ( "x is not an expected value" );
}
```

Lead Image © saracorso, 123rf.com

```
    print ( "3" )
  case _:
    print ( "Not 1, 2 or 3" )
```

The match x says "here's the value to find." Unlike C, the initial value passed in is not in parentheses, just the case values for comparison. Like anything else in Python, code indented under the case is the block that will be executed, and once a block is executed, the match happens without the need for a break statement.

The case _ statement is Python's default case, and if used, it must be the last case to appear in the match. If none of the other case blocks execute, the case _ block executes before the match completes.

But Wait, There's More!

These examples just scratch the surface of what match can do: It also can check variable types, shapes, and variable definitions. The example in Listing 2 [4] converts two- and three-dimensional points.

The two classes Point3D and Point2D define three- and two-dimensional points, respectively. The 3D example defines x, y, and z, and the __init__ function accepts these arguments and assigns the incoming variables to their class equivalents.

The __match_args__ in lines 5 and 18 are part of the new proposal. Usually, it will be generated automatically, but because I'm working with an alpha release (see the"Alpha Version" box), I had to define it myself. Here, match finds positional arguments for comparison. Line 5 proposes a tuple of strings that line up with the argument names that __init__ is collecting. Point2D is identical, except all references to z have been removed, and it does not have a print method.

Just in Case

Lines 24-33 define make3Dpoint, where the match statement resides, in this case to pt. The first case statement on line 26 is looking for two variables in a tuple. If a tuple with two elements is received, this case matches, and x and y are assigned to the variables of the tuple. The function closes by returning a Point3D instance, passing x and y, along with 0, because a z value was not received.

The second case statement on line 28 works the same way, except it matches a three-member tuple. The third case statement on line 30 begins to get even more interesting. Here, the case matches if pt

Alpha Version

The examples in this article are based on an alpha release of Python [5]. I specifically installed this version to work with the new match syntax, but it *should not* be used in production. If you want to install an alpha version for testing on Ubuntu, use the following steps (for other versions, consult your package manager):

```
sudo add-apt-repository §§
  ppa:deadsnakes/ppa
sudo apt update
sudo apt install python3.10
```

When you add the repository, you'll be prompted for your password. The update command updates the package list and includes options from the Python repository added in the previous line. Answer *Yes* to the final command and the alpha version is installed.

You can find the new version of Python by typing whereis python3.10. I did not add the alpha version to my path, so I had to run it directly with /usr/bin/python3.10.

Alpha versions change daily, might not be fully functional, and could have features changed or removed completely before release, so code at your own risk.

Listing 2: example1.py

```
01 class Point3D:
02     x = 0
03     y = 0
04     z = 0
05     __match_args__ = ( "x" , "y" , "z" )
06
07     def __init__ ( self , x , y , z ):
08         self.x = x
09         self.y = y
10         self.z = z
11
12     def print ( self ):
13         print ( "3D Point ( {0} , {1} , {2} )".format (
           self.x , self.y , self.z ) )
14
15 class Point2D:
16     x = 0
17     y = 0
18     __match_args__ = ( "x" , "y" )
19
20     def __init__ ( self , x=None , y=None ):
21         self.x = x
22         self.y = y
23
24 def make3Dpoint ( pt ):
25     match pt:
26         case ( x , y ):
27             return Point3D ( x , y , 0 )
28         case ( x , y , z ):
29             return Point3D ( x , y , z )
30         case Point2D ( x , y ):
31             return Point3D ( x , y , 0 )
32         case Point3D ( _ , _ , _ ):
33             return pt
34
35 pointList = list()
36 pointList.append ( ( 2 , 3 ) )
37 pointList.append ( ( 2 , 3 , 4 ) )
38 pointList.append ( Point2D ( 2 , 3 ) )
39 pointList.append ( Point3D ( 2 , 3 , 4 ) )
40
41 for pt in pointList:
42     threeD = make3Dpoint ( pt )
43     threeD.print()
```

is an instance of `Point2D` *and* `x` and `y` are defined. In this case, `x` and `y` will be assigned to their values from the instance of `Point2D` that is currently being evaluated; then, a `Point3D` is created with the existing `x` and `y`, with `z` as `0`. This `x` and `y` mapping derives from `__match_args__`.

The final `case` statement checks for an instance of `Point3D` and has three positional arguments. An underscore (_) is a `match` wildcard. In that case, it has already been handed a `Point3D`, so it just returns it unmodified.

Line 35 defines `pointList` as a `list`, and then lines 36-39 `append` different sized tuples and class instances to the list of things to be processed. Line 41 loops over `pointList`; line 42 calls `make3Dpoint` which has the `match` statement; and line 43 calls the `print` method on the resulting 3D point (Figure 1).

Nine lines of code have normalized four different styles of input and received uniform objects. With this new syntax, you can take all sorts of input and pass them through a single function that returns consistent objects for further work.

Guards

The `match` statement also includes an additional check called a guard, which is a conditional statement that can further evaluate the input being checked in a `case`. Consider the "guess a number" game (Listing 3). Here, the program chooses a number between 0 and 100, and you must figure it out in as few guesses as possible.

The code begins with an `import` of the `random` library and then creates the random integer `target` between 0 and 100 and initializes a couple of other variables: `guessCount`, the number of guesses that have been made, and `gameOver`, which indicates whether the game is finished.

Guesses

The `checkGuess` function (lines 7-23) contains the `match` statement, but first, lines 8-10 use the `global` keyword to bring in the variables defined earlier so they are accessible inside the function without passing them in each time.

Line 12 starts the `match` on `guess`, which is the player's guess. Notice that all of the `case` statements use `i` instead of `guess`, and because `match` includes variable assignment, `guess` becomes `i`.

You'll also notice that each `case` checks `i` but then is followed by `if`. These `if` statements are called *guards*, and they create extra conditions for the item to match. The first `case` (line 13) checks whether `i` is less than 0 or greater than 100. If so, a *Guess out of range* message is displayed.

The second (line 15) and third (line 18) `case` statements determine whether the guess is too high or too low, display an appropriate message, increment `guessCount`, and exit the `match`.

The last `case` (line 21) checks to see whether the correct number has been guessed. If so, it prints the *Got it!* message and sets `gameOver` to `True` to exit the main loop.

Line 25 sets up the main loop (Figure 2), watching for `gameOver` to become `True`. Until then, it prints the number of guesses (line 26), asks for a guess (line 27), and checks it with the `checkGuess` function (line 28).

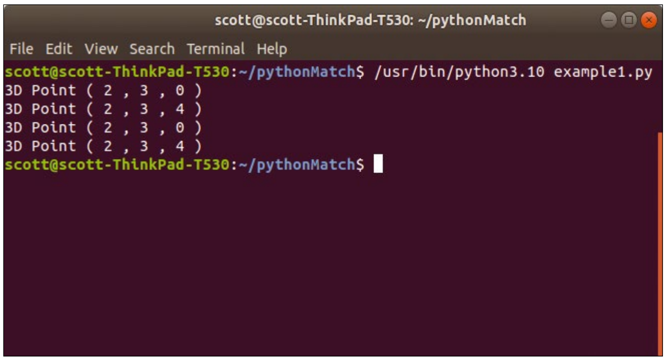
Listing 3: numberGuess.py

```
01 import random
02
03 target = random.randint ( 0 , 100 )
04 guessCount = 0
05 gameOver = False
06
07 def checkGuess ( guess ):
08     global target
09     global guessCount
10     global gameOver
11
12     match guess:
13         case ( i ) if i < 0 or i > 100:
14             print ( "Guess out of range, game is from 0 to 100" )
15         case ( i ) if i < target:
16             print ( "Higher!" )
17             guessCount += 1
18         case ( i ) if i > target:
19             print ( "Lower!" )
20             guessCount += 1
21         case ( i ) if i == target:
22             print ( "Got it!" )
23             gameOver = True
24
25 while gameOver == False:
26     print ( "You've guessed {0} times".format ( guessCount ) )
27     guess = int ( input ( "Your guess:" ) )
28     checkGuess ( guess )
```

Card Game

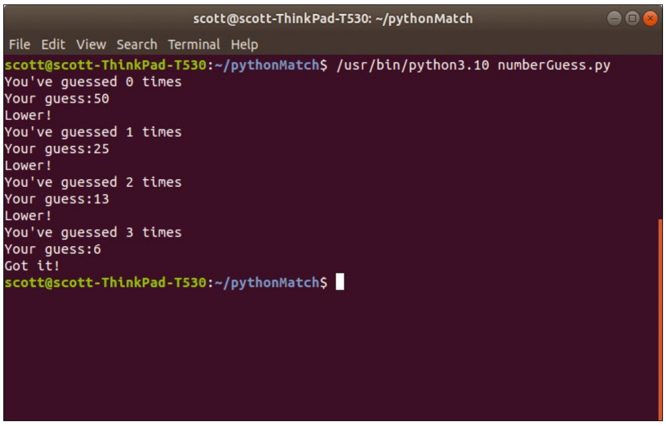
The next example is a card game that brings together everything discussed so far (Listing 4). The hand of cards is defined in the program as the ace of spades, jack of clubs, 2 and 9 of hearts, and 5, 6, and 9 of diamonds (lines 22-28).

To begin, the program imports `dataclass`, which is a helper class that handles some basic definitions. Adding the `@dataclass`


scott@scott-ThinkPad-T530: ~/pythonMatch
File Edit View Search Terminal Help

```
scott@scott-ThinkPad-T530:~/pythonMatch$ /usr/bin/python3.10 example1.py
3D Point ( 2 , 3 , 0 )
3D Point ( 2 , 3 , 4 )
3D Point ( 2 , 3 , 0 )
3D Point ( 2 , 3 , 4 )
scott@scott-ThinkPad-T530:~/pythonMatch$
```

Figure 1: The output of `example1.py`. Note how all of the points are of the same type and align nicely without any missing parameters.


scott@scott-ThinkPad-T530: ~/pythonMatch
File Edit View Search Terminal Help

```
scott@scott-ThinkPad-T530:~/pythonMatch$ /usr/bin/python3.10 numberGuess.py
You've guessed 0 times
Your guess:50
Lower!
You've guessed 1 times
Your guess:25
Lower!
You've guessed 2 times
Your guess:13
Lower!
You've guessed 3 times
Your guess:6
Got it!
scott@scott-ThinkPad-T530:~/pythonMatch$
```

Figure 2: A game of guess the number.

decorator before the definition (line 3) generates the __init__ function automatically and assigns attributes of the same name to class variables. Therefore, in lines 22-28, where a hand of cards is defined, the values are automatically assigned to rank and suit. The match already knows how to work with a dataclass, so __match_args__ does not have to be defined, as in an earlier example. The variables rank and suit are defined in lines 5 and 10.

Card Names

The cardName function uses a match (line 13) to generate a human-readable string of the card name. Line 14 checks to see whether both rank and suit are defined in a case statement. If both are present, the program checks for a numerical card with the guard if rank <= 10. If it matches, the print outputs the string {0} of {1}, where format replaces 0 with rank and 1 with suit.

The rest of the case statements (lines 16-19) check for a particular rank by specifying =11, =12, and so on. The suit = _ element says that suit must be defined in the instance, but it doesn't matter what value it holds. Each case then prints the name of the card and its suit. Note that you can check for a particular value just by specifying it in a case.

Each of the remaining code blocks uses different combinations of case and guards to extract different cards from the player's hand.

Card Matching

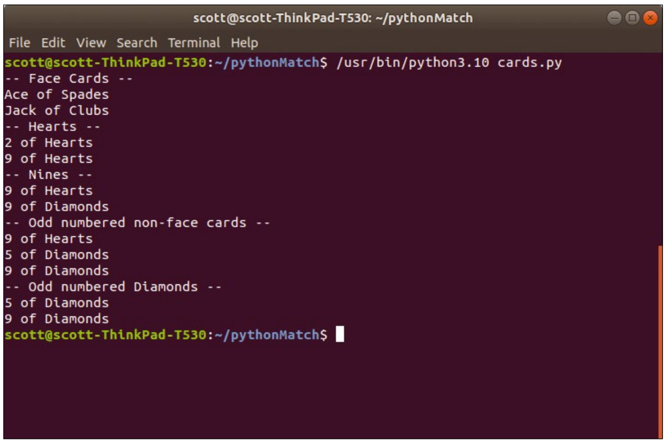
The match in lines 32-34 checks that a rank and suit is present and then uses the guard if rank >= 11 to return only cards ranked 11 or higher.

The -- Hearts -- print block (lines 36-40) only has to specify suit = "Hearts" in the case statement for all of the hearts in the hand. The -- Nines -- case (lines 42-46) only specifies rank = 9. Optionally it could specify suit with no condition to make sure it is defined.

The non-face cards section (lines 48-52) is a complex condition, but it is still just a single line in the guard. It requests rank and then uses rank % 2 to check for a remainder. If true, then the card is odd. It also checks rank <= 10 so that only non-face cards are returned.

Listing 4: cards.py

```
01 from dataclasses import dataclass
02
03 @dataclass
04 class playingCard:
05    rank: int
06    # 11 - Jack
07    # 12 - Queen
08    # 13 - King
09    # 14 - Ace
10    suit: str
11
12 def cardName ( card ):
13    match card:
14       case playingCard ( rank , suit ) if rank <= 10:
15          print ( "{0} of {1}".format ( rank , suit ) )
16       case playingCard ( rank = 11 , suit = _ ): print ( "Jack of " + suit )
17       case playingCard ( rank = 12 , suit = _ ): print ( "Queen of " + suit )
18       case playingCard ( rank = 13 , suit = _ ): print ( "King of " + suit )
19       case playingCard ( rank = 14 , suit = _ ): print ( "Ace of " + suit )
20
21 deck = list()
22 deck.append ( playingCard ( 14 , "Spades" ) )
23 deck.append ( playingCard ( 11 , "Clubs" ) )
24 deck.append ( playingCard ( 2 , "Hearts" ) )
25 deck.append ( playingCard ( 9 , "Hearts" ) )
26 deck.append ( playingCard ( 5 , "Diamonds" ) )
27 deck.append ( playingCard ( 6 , "Diamonds" ) )
28 deck.append ( playingCard ( 9 , "Diamonds" ) )
29
30 print ( "-- Face Cards --" )
31 for card in deck:
32    match card:
33       case playingCard ( rank , suit ) if rank >= 11:
34          cardName ( card )
35
36 print ( "-- Hearts -- " )
37 for card in deck:
38    match card:
39       case playingCard ( rank , suit = "Hearts" ):
40          cardName ( card )
41
42 print ( "-- Nines --" )
43 for card in deck:
44    match card:
45       case playingCard ( rank = 9 ):
46          cardName ( card )
47
48 print ( "-- Odd numbered non-face cards --" )
49 for card in deck:
50    match card:
51       case playingCard ( rank ) if rank % 2 == 1 and rank <= 10:
52          cardName ( card )
53
54 print ( "-- Odd numbered Diamonds --" )
55 for card in deck:
56    match card:
57       case playingCard ( rank , suit = "Diamonds" )
        if rank % 2 == 1 and rank <= 10:
58          cardName ( card )
```



```
scott@scott-ThinkPad-T530: ~/pythonMatch
File  Edit  View  Search  Terminal  Help
scott@scott-ThinkPad-T530:~/pythonMatch$ /usr/bin/python3.10 cards.py
-- Face Cards --
Ace of Spades
Jack of Clubs
-- Hearts --
2 of Hearts
9 of Hearts
-- Nines --
9 of Hearts
9 of Diamonds
-- Odd numbered non-face cards --
9 of Hearts
5 of Diamonds
9 of Diamonds
-- Odd numbered Diamonds --
5 of Diamonds
9 of Diamonds
scott@scott-ThinkPad-T530:~/pythonMatch$ 
```

Figure 3: The output of *cards.py*, demonstrating different combinations of `case` and guard statements.

Finally, the program specifies the `suit` so only diamonds are matched (lines 54-58). The guard `if rank % 2 == 1 and rank <= 10` checks for an odd number as described before and then eliminates all cards that are not numbers (Figure 3).

Conclusion

The `match` statement included with Python 3.10 is slated to be release in October 2021. As you can see from the few examples I explored here, it can streamline a number of tasks and eliminate many, many `if`/`elif` blocks. If you start learning this new syntax now, you'll be well on your way to even more functional Python code as soon as this release becomes official! ■■■

Author

Scott Sumner is the Assistant Manager of the Charlie Noble Planetarium at the Fort Worth Museum of Science and History. He enjoys using Python to solve as many problems as possible.

Info

[1] Python: *http://www.python.org*

[2] Python PEP 622 original match proposal: *https://www.python.org/dev/peps/pep-0622/*

[3] Python PEP 634 (replaces PEP 622): *https://www.python.org/dev/peps/pep-0634/*

[4] Code for this article: *ftp://ftp.linux-magazine.com/pub/listings/linux-magazine.com/249/*

[5] Python 3.10 alpha: *https://www.python.org/downloads/release/python-3100a6/*

Public Money Public Code

Modernising Public Infrastructure with Free Software

fsfe Free Software Foundation Europe

Learn More: https://publiccode.eu/

Freehand writing and drawing

VIRTUAL WHITEBOARD

Xournal brings the whiteboard to the PC to aid in all of your online learning activities. *By Erik Bärwaldt*

Over the past year, an increase in homeschooling has given an unprecedented boost to learning management, communication systems, and curriculum-based software. When using these online learning tools, a virtual whiteboard can help enhance learning.

Xournal [1], a note-taking, sketching, and journaling app, can turn your device into a whiteboard, letting you visualize content via a stylus or tablet. Xournal's workspace mimics a notebook page and offers a variety of tools, including a shape recognizer that recognizes geometric shapes as they are drawn. Not only can you take notes and make simple drawings, but you can also use Xournal for screencasts and real-time screen sharing during a video conference.

Installation

Xournal can be found in the repositories of all major distributions. For numerous Linux derivatives, there are packages for both 32- and 64-bit machines. In addition, the software also runs on various BSD derivatives.

After installation, you start Xournal conveniently via a launcher in your desktop environment's menu structure. Besides a conventional menubar at the top of the program window, you will find a two-line buttonbar for quick selec-

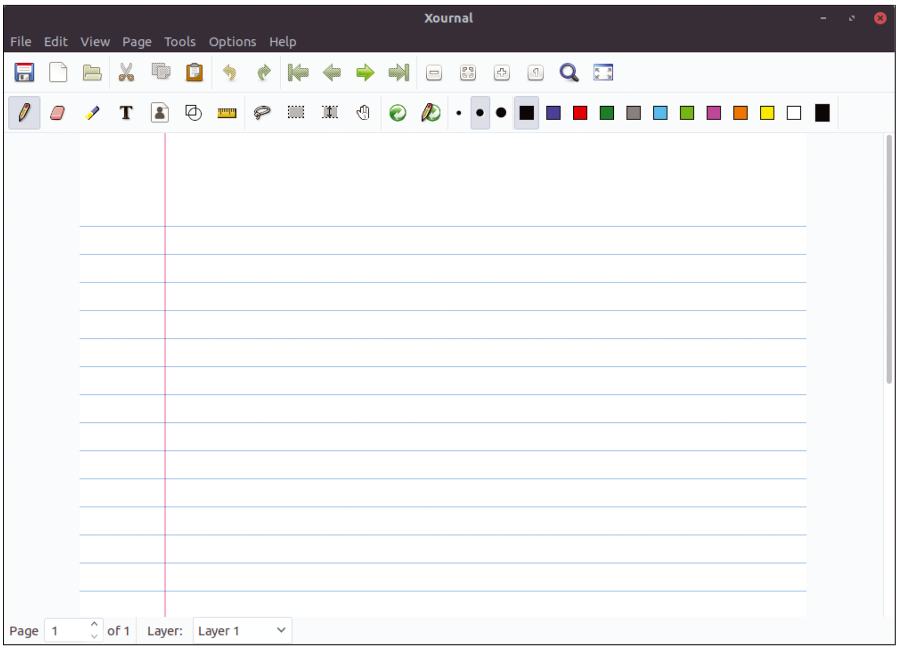


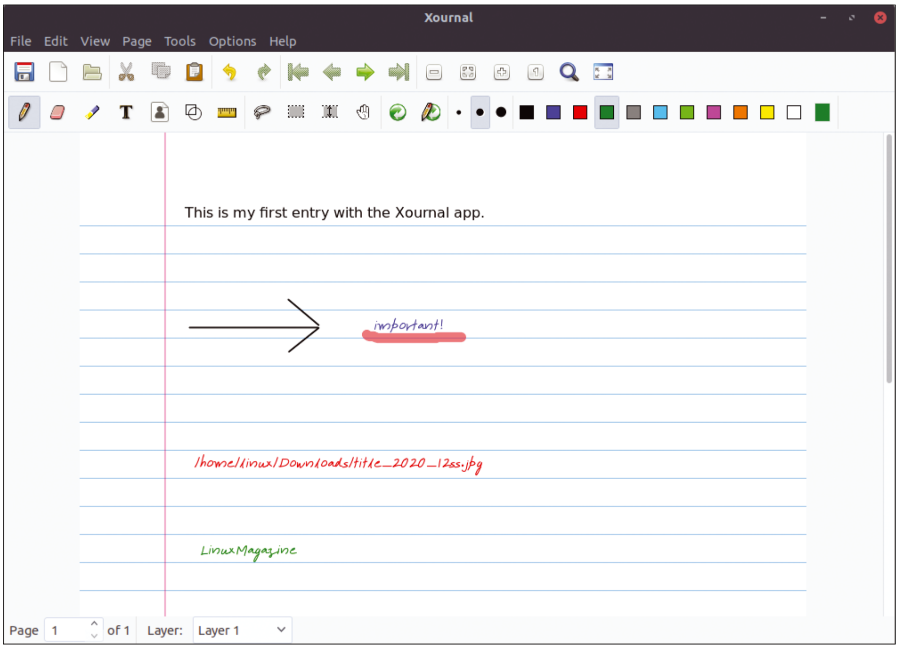
Figure 1: The Xournal workspace mimics a school exercise book.

Lead Image © Galina Peshkova, 123RF.com

tion of drawing tools and attributes. The workspace below looks like lined notebook paper, which helps with placing your content (Figure 1).

To get started, go to *Page* | *Paper Style* to specify the workspace format.

In addition to the default lined paper style, you can also choose graph, ruled, or blank paper styles. Depending on your printing needs, you can change the default US standard paper size (say, to DIN A4) via *Page* | *Paper Size*. To permanently save your settings, check the *Apply to all pages* option in each dialog box.


Xournal
File Edit View Page Tools Options Help

This is my first entry with the Xournal app.

important!

/home/linux/Downloads/title_2020_12ss.jpg

LinuxMagazine

Page 1 of 1 Layer: Layer 1

Figure 2: Xournal offers several tools for customizing your content.

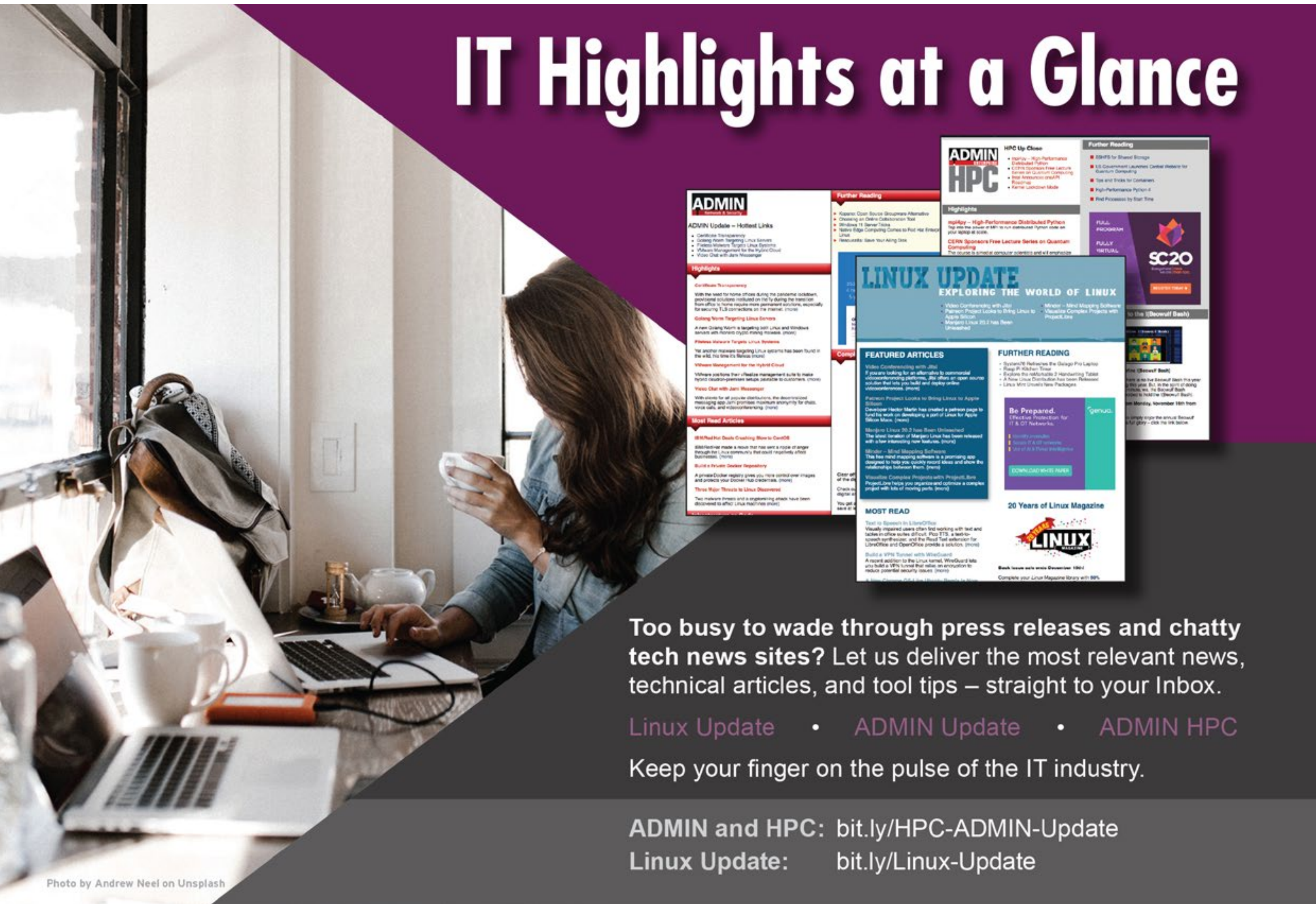
Tools

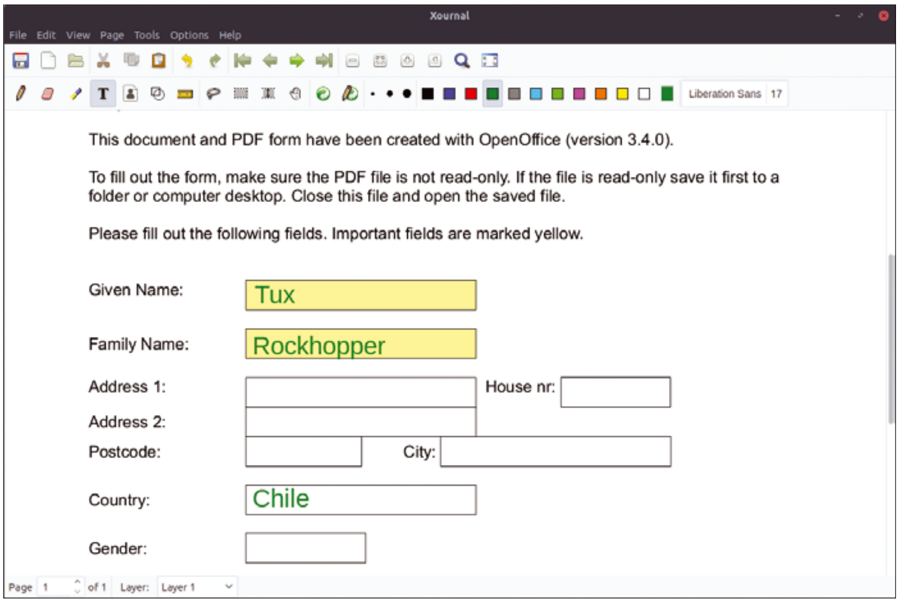
Xournal offers numerous tools for entering content (Figure 2). In the buttonbar, you will find the pen, eraser, highlighter, text, image, ruler, and shape recognizer tools. The shape recognizer tool recognizes geometric shapes as you draw them and converts the hand-drawn lines into clean lines.

When using the text tool, first set a font via *Tools* | *Font* (Xournal uses its own fonts). To define the font weight, go to *Tools* | *Pen Options*. You can also set the eraser thickness in *Tools* | *Eraser Options*.

To select the color used by a tool, you can choose from the limited color options in the lower buttonbar or go to *Tools* | *Colors* for more options.

To select a portion of the workspace, you can use *Select Rectangle* and *Select Region* options. You can then modify the content attributes in the selected area. This comes in handy if you want to




Xournal
File Edit View Page Tools Options Help
Liberation Sans 17
Page 1 of 1 Layer: Layer 1

This document and PDF form have been created with OpenOffice (version 3.4.0).

To fill out the form, make sure the PDF file is not read-only. If the file is read-only save it first to a folder or computer desktop. Close this file and open the saved file.

Please fill out the following fields. Important fields are marked yellow.

Given Name: Tux

Family Name: Rockhopper

Address 1: House nr:

Address 2:

Postcode: City:

Country: Chile

Gender:

Figure 3: Fill out forms directly in Xournal.

quickly modify the font color in a longer text passage, for example.

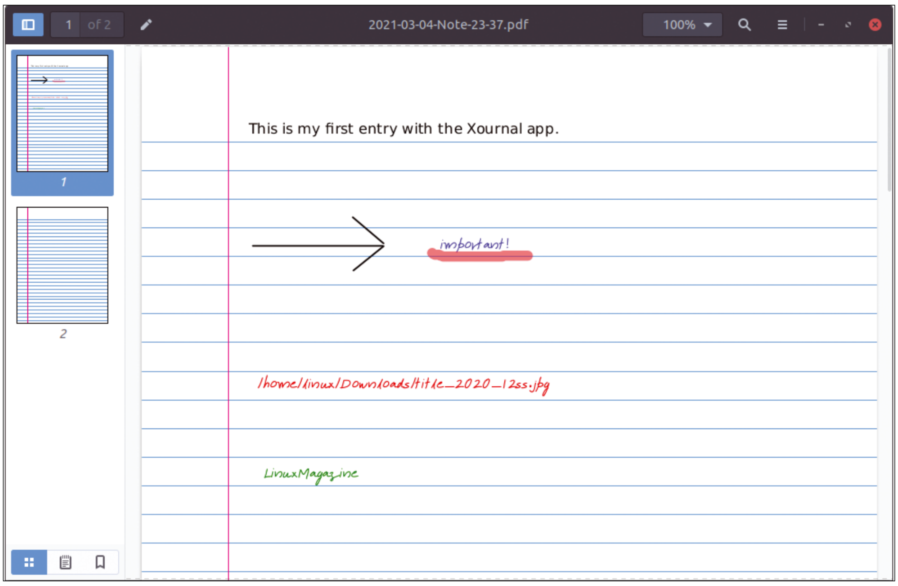
In principle, Xournal can use all available input devices to generate content. In addition to the keyboard for text input, you can use the mouse and, if available, a graphics tablet.

Xournal also can be used with the stylus and touchscreen found in some modern notebooks or older convertible systems. Under *Options* | *Pen and Touch*, you can set the stylus's pressure sensitivity or choose to operate the touchscreen without a stylus (using your fingers instead). In addition, you can specify which keys are used to quickly toggle between the input options.

Page by Page

To allow content to extend beyond the visible workspace, Xournal supports page-by-page content capture and modification. You can insert a new page at any point before or after the visible page with *Page* | *New Page Before* or *New Page After*. To insert the new page at the very end of the workspace, use the *New Page at End* option.

Similarly, you can delete a page via *Page* | *Delete Page*; you can only delete the page currently shown in the workspace. Xournal does not let you select multiple pages for deletion in one step. To scroll through the individual pages, either use the page count at the bottom, which you can use to jump across multiple pages if necessary, or navigate in the upper buttonbar using the arrow keys.



This is my first entry with the Xournal app.

important!

/home/linux/Downloads/title_2020_12ss.jpg

LinuxMagazine

Figure 4: Exporting to PDF format makes the Xournal files usable in other applications.

Forms

Given its ability to read and edit PDF files, Xournal is also great for filling out forms digitally. Xournal places a second layer, where you make your entries, on top of the actual form. First, load the document into Xournal via *File* | *Annotate PDF*. Then fill out the form (Figure 3) with all the design and font elements at your disposal.

When you've finished filling out the form, either save the entire document in Xournal's own file format using *File* | *Save* or save it as a PDF file in *File* | *Export as PDF*. Xournal merges the two layers, which transfers the manually filled areas to the new document.

Since Xournal uses its own format for saving content, you can only open, process, and print these files in Xournal. If you need to further process the files in a third-party application, choose the PDF option, which then lets you view and edit in any PDF editor. Xournal always stores the files as complete pages (Figure 4).

Conclusions

Whether you are using Xournal in a virtual classroom or a team meeting, Xournal can help you communicate your ideas. With freehand drawing support, you can easily turn Xournal into a virtual whiteboard to supplement other online learning formats. With Xournal's PDF export option, you can also make your content available to other participants at the push of a button. You will definitely want to bring Xournal along to your next virtual meeting. ■■■

Info

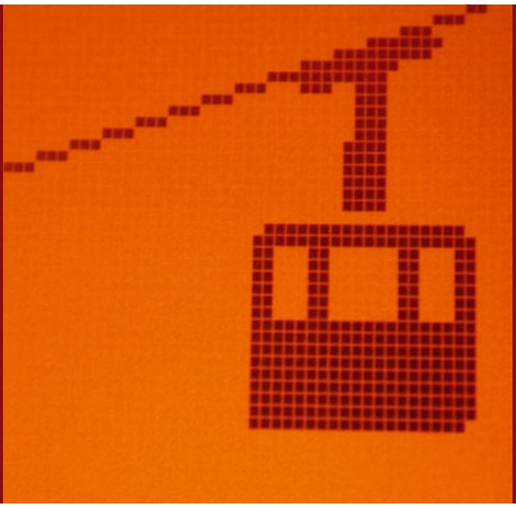
[1] Xournal: *http://xournal.sourceforge.net*

Author

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

20 YEARS
LINUX
MAGAZINE
Searchable Linux Archive: 21,000 Pages of Practical Know-How
20 YEARS OF LINUX! Issues 1-239
All the Tricks
All the Hacks
All the Apps
THE COMPLETE LINUX MAGAZINE ARCHIVE
20 YEARS

WEBCAMS AND LINUX
SPEED UP YOUR SYSTEM
IoT on the Cheap
BUILD AN IRC BOT
INTRUSION DETECTION
Systemd
SPOTLIGHT ON SMALL DISTROS
AUTOMATION TRICKS
BITKEY
SECURING LINUX
PRIVACY



MakerSpace

Automate your web logins

Log In and Go

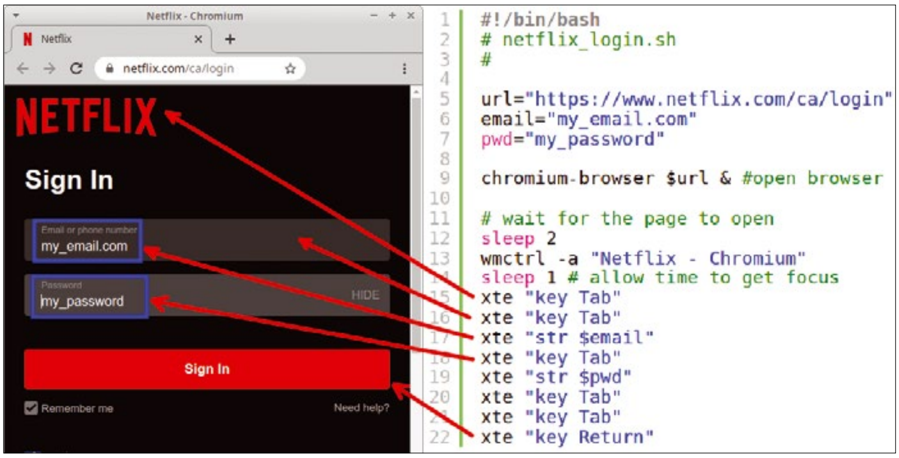
Automated web logins with command-line tools and Selenium ensure you don't miss scheduling an activity.

By Pete Metcalfe

During the COVID-19 lockdown, many activities like pools, gyms, and golf courses required people to sign in to websites before they could access these activities. These precautions helped to maintain a safe environment; however, the booking process was awkward, and it was easy to miss an activity if you weren't signed up early enough.

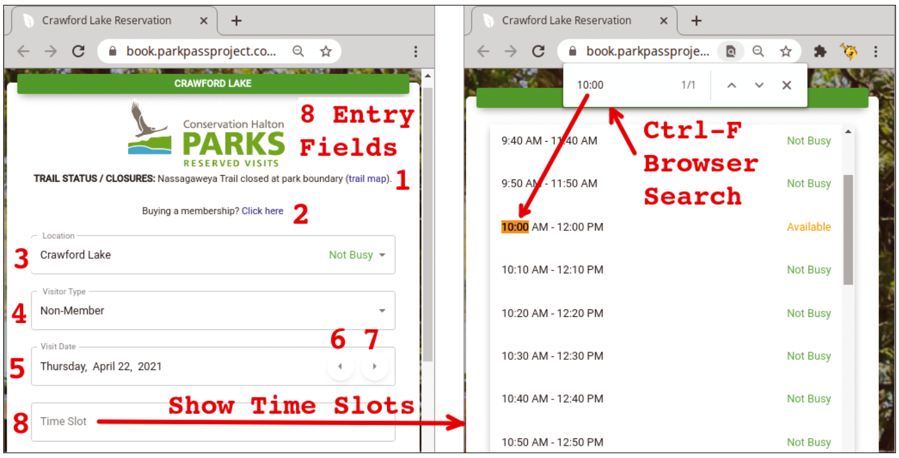
Luckily, some great Linux tools can automate web logins. In this article, I share two techniques to create automated web logins. The first technique uses command-line tools like `xte` and `xdotool`. This approach allows simple Bash scripts to replicate how you would use keystrokes to access web pages.

The second technique uses the Selenium Python library. The Selenium API allows you to tackle more complex projects by giving you access to the full Document Object Model (DOM) of a web page.



```
#!/bin/bash
# netflix_login.sh
#

url="https://www.netflix.com/ca/login"
email="my_email.com"
pwd="my_password"

chromium-browser $url & #open browser

# wait for the page to open
sleep 2
wmctrl -a "Netflix - Chromium"
sleep 1 # allow time to get focus
xte "key Tab"
xte "key Tab"
xte "str $email"
xte "key Tab"
xte "str $pwd"
xte "key Tab"
xte "key Tab"
xte "key Return"
```

Figure 1: Netflix sign in with the `xte` command.



TRAIL STATUS / CLOSURES: Nassagaweya Trail closed at park boundary (trail map).

Buying a membership? Click here

Location: Crawford Lake — Not Busy

Visitor Type: Non-Member

Visit Date: Thursday, April 22, 2021

Time Slot

9:40 AM - 11:40 AM	Not Busy
9:50 AM - 11:50 AM	Not Busy
10:00 AM - 12:00 PM	Available
10:10 AM - 12:10 PM	Not Busy
10:20 AM - 12:20 PM	Not Busy
10:30 AM - 12:30 PM	Not Busy
10:40 AM - 12:40 PM	Not Busy
10:50 AM - 12:50 PM	Not Busy

Figure 2: Automating a park booking with a time search.

Keyboard Simulation

The most popular choices for keyboard and mouse simulation are the *xautomation* package [1] and the `xdotool` utility [2]. The `xdotool` utility is feature-rich, with special functions for desktop and window functions. The `xte` tool, a part of *xautomation*, is a little simpler, focusing entirely on keyboard and mouse simulation.

The `wmctrl` [3] utility is also very useful to help you determine which windows are open on your desktop, and it can also set the active window with a text substring.

In Ubuntu, enter

```
sudo apt-get install ⮐
  xautomation xdotool wmctrl
```

to install the *xautomation* package and the `xdotool` and `wmctrl` utilities.

Log In with xte

With the `xte` utility, you can send a single keyboard character or strings of characters. A Bash script that uses `xte`

Lead Image © thotti, photocase.com

commands can emulate your actions to log in manually to a web page.

Typically people use the mouse on web pages, which is quite different from logging in 100 percent with the keyboard. Web pages often have a number of clickable items before the main form entry area, so it is important to step through and document the login procedure manually. A good simple example is to try and log in to Netflix (Figure 1).

The Bash script in Listing 1 uses `xte` to automate the Netflix sign-in. This script opens a Chrome browser page (line 10) and then sets the focus to this page (line 12). Next, it sends the correct tab, text, and return key sequences (lines 15-22).

Setting the window focus can be tricky if you have a number of windows open. The command `wmctrl -l` lists all open windows, and the command

```
wmctrl -a '<some title info>'
```

sets the mouse and keyboard focus to a specific window from a substring of the window title.

Book with xdotool

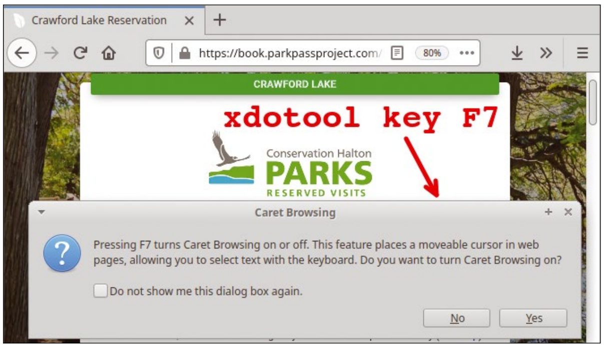
The `xdotool` syntax also sends keystrokes and text and is very similar to `xte`, but with a few extra features. A park booking example (Figure 2) is a bit more complex, because a booking time needs to be selected from a list. For this project, the automation script needs to manage eight entry fields (to keep things simple, I'll pass the date in the URL) and select a time slot.

Neither the `xte` nor `xdotool` utility supports a search text function. A simple workaround is to use the web browser's search function. By enabling caret (text cursor) navigation, it's possible to move the active cursor location according to the browser's search results.

The caret dialog is shown by pressing F7 (Figure 3). It's important to note that the caret enable/cancel and Yes/No buttons can vary between browsers.

The Bash script in Listing 2 uses the browser's search dialog to find and select a 10:00am time slot for a park. One of the first steps is to enable caret navigation (lines 12-13).

A useful feature of `xdotool` is the repeat with a delay option (lines 17-18). In this script, I used this feature to tab eight times to get to the *Time Slot* field. A Ctrl+F keystroke opens the browser search dialog (line 24). Next, the `xdotool type` option passes in the `'10:00'` time string (line 25). The final step is to close the search dialog and hit Return to select the *10:00 AM – 12:00 PM* time slot (line 28).



Pressing F7 turns Caret Browsing on or off. This feature places a moveable cursor in web pages, allowing you to select text with the keyboard. Do you want to turn Caret Browsing on?

☐ Do not show me this dialog box again.

Figure 3: Improve keyboard navigation with caret browsing.

Script Limitations

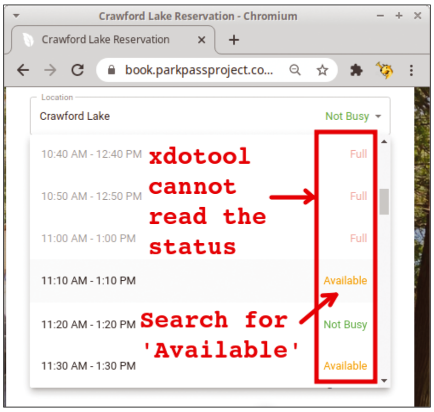
The `xdotool` and `xte` utilities are great for simple web page automation when the

Listing 1: Netflix Sign-In

```
01 #!/bin/bash
02 # netflix_login.sh - script logs into Netflix
03 #
04
05 url="https://www.netflix.com/ca/login"
06 email="my_email.com"
07 pwd="my_password"
08
09 # open browser to wait for the page to open, then set
    focus to it
10 chromium-browser $url &
11 sleep 2
12 wmctrl -a "Netflix - Chromium"
13
14 sleep 1 # allow time to get focus before sending keys
15 xte "key Tab"
16 xte "key Tab"
17 xte "str $email"
18 xte "key Tab"
19 xte "str $pwd"
20 xte "key Tab"
21 xte "key Tab"
22 xte "key Return"
23
24 echo "Netflix Login Done..."
```

Listing 2: Book a Park Visit

```
01 #!/bin/bash
02 # book10am.sh - make a 10:00 park booking
03 #
04 sdate="startDate=2021-04-23" #adjust the date
05 url="https://book.parkpassproject.com/book?inventoryGroup
        =1554186518&&inventory=1229284276&$sdate"
06
07 chromium-browser $url & #open browser to park booking page
08 sleep 5 # wait for browser to come up
09 wmctrl -a "Chromium"
10 sleep 2
11 # Turn on caret browsing
12 xdotool key F7
13 xdotool key Return
14 sleep 1
15
16 # tab to 'Time Slot' area
17 tabcnt=8
18 xdotool key --repeat $tabcnt --delay 100 Tab
19
20 xdotool key Return
21 sleep 1
22
23 # Search for 10:00 time and select it
24 xdotool key ctrl+f
25 xdotool type '10:00'
26 xdotool key Return
27 # Close find dialog and select time
28 xdotool key Tab Tab Tab Return Return
29
30 echo "Park Time Booking Complete"
```



Crawford Lake Reservation - Chromium

Crawford Lake Reservation

book.parkpassproject.co...

Location
Crawford Lake Not Busy

10:40 AM - 12:40 PM Full
10:50 AM - 12:50 PM Full
11:00 AM - 1:00 PM Full
11:10 AM - 1:10 PM Available
11:20 AM - 1:20 PM Not Busy
11:30 AM - 1:30 PM Available

xdotool cannot read the status

Search for 'Available'

Figure 4: Limitations with keystroke automation.

HTML form items are sequential and no special decision making is required. Unfortunately, I found that when I tried to book a park time on the weekend, I started to see some limitations (Figure 4). During busy times, if I tried to book by time, `xte` and `xdotool` could not determine whether the time slot was taken. A simple workaround would be to search for the first *Available* or *Not Busy* slot, but this doesn't allow you to pick times you like.

For projects that require some logic (like choosing a good time from a list of times), Selenium with Python is an excellent fit.

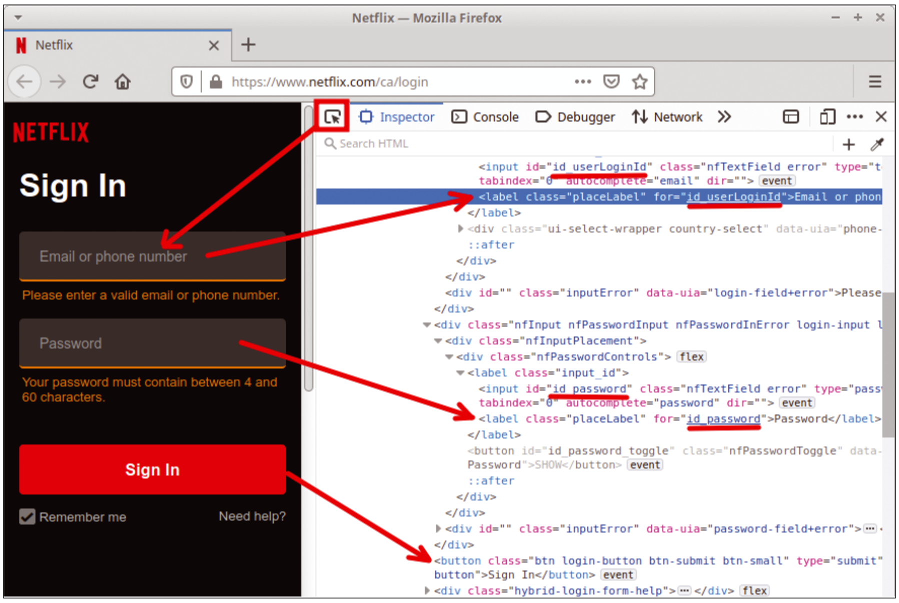
Selenium with Python

Selenium [4] is a portable framework for testing web applications, with client-server tools and an IDE. The Selenium WebDriver component (available for Firefox, Google Chrome, Internet Explorer, Safari, Opera, and Edge) sends commands from client APIs directly to a browser. Client APIs are available for C#, Go, Java, JavaScript, PHP, Python, and Ruby. The Selenium Downloads page [5] has details on installation of the WebDriver scripts.

To install the Linux 32-bit Selenium driver (`geckodriver`) for Firefox, enter:

```
wget https://github.com/mozilla/↩
  geckodriver/releases/download/↩
  v0.29.1/geckodriver-v0.29.1-↩
  linux32.tar.gz
tar -xvzf ↩
  geckodriver-v0.24.0-linux32.tar.gz
chmod +x geckodriver
sudo mv geckodriver /usr/local/bin
```

To install the Selenium library for Python, enter:



Netflix — Mozilla Firefox

Netflix

https://www.netflix.com/ca/login

NETFLIX

Sign In

Email or phone number

Please enter a valid email or phone number.

Password

Your password must contain between 4 and 60 characters.

Sign In

☑ Remember me Need help?

```
<input id="id_userLoginId" class="nfTextField error" type="t
tabindex="0" autocomplete="email" dir=""> event
<label class="placeLabel" for="id_userLoginId">Email or phon
</label>
<div class="ui-select-wrapper country-select" data-uia="phone-
::after
</div>
</div>
<div id="" class="inputError" data-uia="login-field+error">Please
</div>
<div class="nfInput nfPasswordInput nfPasswordInError login-input l
<div class="nfInputPlacement">
<div class="nfPasswordControls"> flex
<label class="input_id">
<input id="id_password" class="nfTextField error" type="pass
tabindex="0" autocomplete="password" dir=""> event
<label class="placeLabel" for="id_password">Password</label>
</label>
<button id="id_password_toggle" class="nfPasswordToggle" data-
Password">SHOW</button> event
::after
</div>
</div>
<div id="" class="inputError" data-uia="password-field+error">…
</div>
<button class="btn login-button btn-submit btn-small" type="submit"
button">Sign In</button> event
<div class="hybrid-login-form-help">…</div> flex
```

Figure 5: Finding input field IDs with Inspector.

```
pip install selenium
```

The big difference between the `xte` or `xdotool` utility and Selenium is that Selenium can access the HTML code of the selected web page directly.

Log In with Selenium and Python

As for `xte` and `xdotool`, you need to do some background manual work before writing the script. Once the required web page is open, you can use the Web Developer Inspector tool to examine HTML code. To access the Inspector, Select *Tools* | *Web Developer* | *Inspector* from the top menubar or use the shortcut Ctrl + Shift + C.

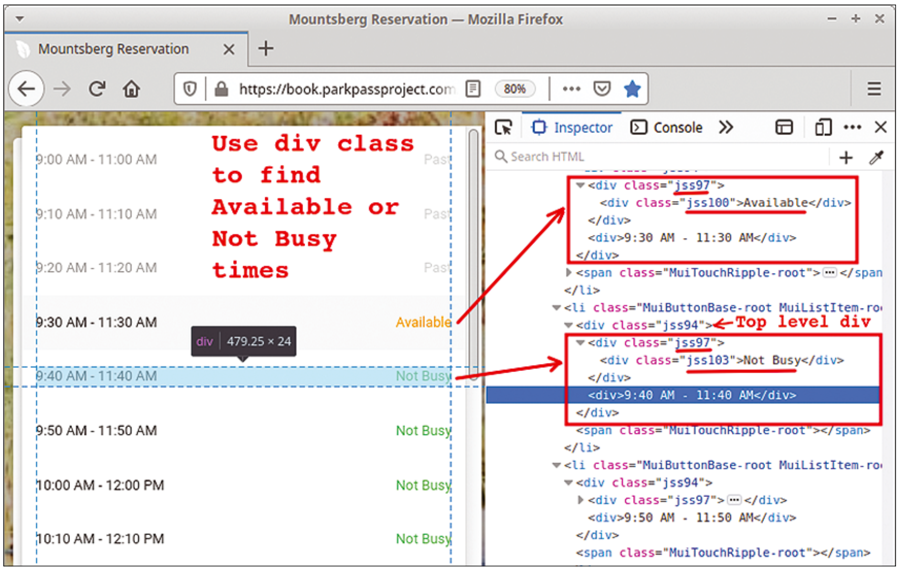
For the Netflix sign-in example, the *Email or phone number* and *Password* inputs are needed (Figure 5). When the Inspector is open, items selected on the web page are highlighted in the Inspector pane. In this example, the *Email or phone number* entry uses `id="id_userLoginId"`, and the password entry uses `id="id_password"`. Listing 3 shows the Python code that signs in to Netflix.

When a web page is called, it's important to give the page some time to refresh. The *implicity_wait(10)* call (line 15) waits up to 10 seconds for a Selenium query to complete.

HTML items can be found by either ID (`find_element_by_id()`) or by name

Listing 3: Netflix Sign-In with Selenium

```
01 #
02 # netflix_login.py - automate Netflix Login
03 #
04 from selenium import webdriver
05
06 url="https://www.netflix.com/ca/login"
07 email="my_email.com"
08 pwd="my_password"
09
10 browser = webdriver.Firefox()
11
12 browser.get(url)
13
14 # wait for page to refresh
15 browser.implicitly_wait(10)
16
17 username = browser.find_element_by_id('id_userLoginId')
18 username.send_keys(email)
19
20 password = browser.find_element_by_id('id_password')
21 password.send_keys(pwd)
22
23 password.submit()
24
25 print("Login Complete")
```


Mountsberg Reservation — Mozilla Firefox
Mountsberg Reservation
https://book.parkpassproject.com
80%
Inspector
Console
Search HTML
div 479.25 × 24

```
9:00 AM - 11:00 AM     Past
9:10 AM - 11:10 AM     Past
9:20 AM - 11:20 AM     Past
9:30 AM - 11:30 AM     Available
9:40 AM - 11:40 AM     Not Busy
9:50 AM - 11:50 AM     Not Busy
10:00 AM - 12:00 PM    Not Busy
10:10 AM - 12:10 PM    Not Busy
```

```
    <div class="jss97">
      <div class="jss100">Available</div>
    </div>
    <div>9:30 AM - 11:30 AM</div>
  </div>
  <span class="MuiTouchRipple-root">…</span
</li>
<li class="MuiButtonBase-root MuiListItem-ro
  <div class="jss94">
    <div class="jss97">
      <div class="jss103">Not Busy</div>
    </div>
    <div>9:40 AM - 11:40 AM</div>
  </div>
  <span class="MuiTouchRipple-root"></span>
</li>
<li class="MuiButtonBase-root MuiListItem-ro
  <div class="jss94">
    <div class="jss97">…</div>
    <div>9:50 AM - 11:50 AM</div>
  </div>
  <span class="MuiTouchRipple-root"></span>
```

Use div class to find Available or Not Busy times

Top level div

Figure 6: Getting list details.

(`find_element_by_name()`). A Selenium object needs to be created before initiating any action on it. Line 17 finds and then creates a `username` object from ID `'id_userLoginId'`. The `send_keys()` method is used to pass text strings to `<input>` tags (lines 18 and 21). Calling the `submit()` method on any input object will send all the form data as a request to the web server (line 23).

Selenium Searches

From the earlier park booking example, you saw that `xte` and *xautomation* had some limitations when a variable list of options was presented. Luckily Selenium has a number of functions that can be used for searching HTML tags and text. Like the last example, the first step is to open the web page and inspect the structure manually (Figure 6).

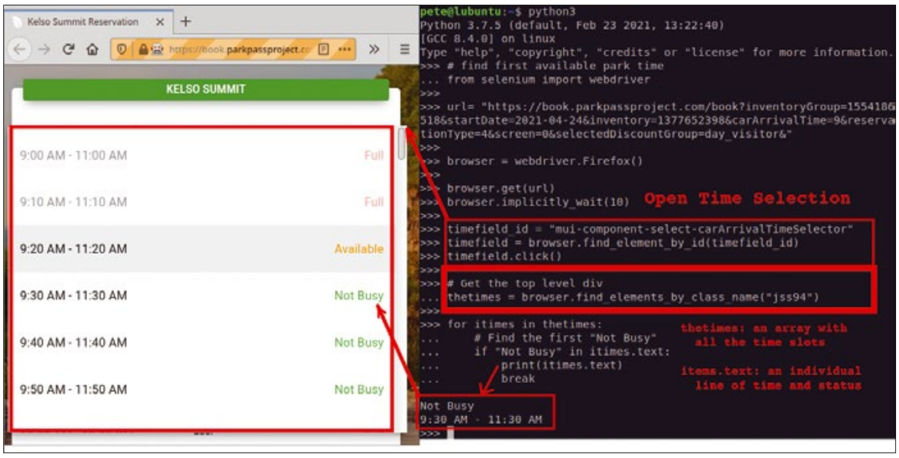
For this example, the Inspector shows that each status entry in the list has a `<div class="jss97">` that could have a *Past*, *Available*, *Not Busy*, or *Full* status. The top-level `<div class="jss94">` has both the times and the status messages. Knowing the top-level `div class` now makes it possible to search for the park's time slots and get the status of each of the times.

Figure 7 shows an example that searches for the first *Not Busy* time slot. As in the earlier `xdotool` example, the time slot list needs to be clicked to open. In Python code, this is done by finding and then clicking on the `timefield` object.

The key piece in this code is:

```
thetimes = ⮒
  find_elements_by_class_name("jss94")
```

This operation will build an array (`thetimes`) of all the time slots with their status messages.



```
9:00 AM - 11:00 AM     Full
9:10 AM - 11:10 AM     Full
9:20 AM - 11:20 AM     Available
9:30 AM - 11:30 AM     Not Busy
9:40 AM - 11:40 AM     Not Busy
9:50 AM - 11:50 AM     Not Busy
```

```
pete@lubuntu:~$ python3
Python 3.7.5 (default, Feb 23 2021, 13:22:40)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> # find first available park time
... from selenium import webdriver
>>>
>>> url= "https://book.parkpassproject.com/book?inventoryGroup=1554186
518&startDate=2021-04-24&inventory=1377652398&carArrivalTime=9&reserva
tionType=4&screen=0&selectedDiscountGroup=day_visitor&"
>>>
>>> browser = webdriver.Firefox()
>>>
>>> browser.get(url)
>>> browser.implicitly_wait(10)
>>>
>>> timefield_id = "mui-component-select-carArrivalTimeSelector"
>>> timefield = browser.find_element_by_id(timefield_id)
>>> timefield.click()
>>>
>>> # Get the top level div
... thetimes = browser.find_elements_by_class_name("jss94")
>>>
>>> for itimes in thetimes:
...     # Find the first "Not Busy"
...     if "Not Busy" in itimes.text:
...         print(itimes.text)
...         break
...
Not Busy
9:30 AM - 11:30 AM
>>>
```

Open Time Selection

thetimes: an array with all the time slots

items.text: an individual line of time and status

Figure 7: Finding the first *Not Busy* time.

Next, a `for` loop can examine each time slot. In this example, the code looks for the first time a time slot is *Not Busy*:

```
# Get the top level div
thetimes = browser.find_elements_by_⮒
           class_name("jss94")

for itimes in thetimes:
    # Find the first "Not Busy"
    if "Not Busy" in itimes.text:
        print(itimes.text)
        itimes.click()
        break
```

Logic could be written for different conditions, like looking for time slots between 9 and 11am, and if none are found, then looking for time slots between 2 and 4pm.

Final Comments

After using the various methods discussed in this article, I found that:

- Often my apps written during off hours would not work during peak times because I had not accounted for the increased peak callup delays.
- The browser search dialog with `xte/xdotool` was extremely useful because it allowed me to jump to specific areas of a web page, rather than tabbing to it.
- Creating apps with `xte` or `xdotool` is considerably easier than using Python with Selenium. I found that some web pages were incredibly complex, and it often took some time to find the required IDs that Selenium needed.
- For large web entry pages, you can always create automated web logins by mixing and matching the `xte/xdotool` utilities and Python.
- Two huge advantages in using Selenium are being able to add some decision-making logic and `implicity_wait()` methods, which wait until the page is ready and is a lot more efficient than putting in a long sleep time. ■■■

Info

[1] xautomation man page: *https://linux.die.net/man/7/xautomation*

[2] xdotool man page: *https://man.cx/xdotool*

[3] wmctrl man page: *https://linux.die.net/man/1/wmctrl*

[4] Selenium: *https://en.wikipedia.org/wiki/Selenium_(software)*

[5] Selenium downloads: *https://www.selenium.dev/downloads/*

FREE DVD!
openSUSE Leap 15.2
JOIN THE LINUX REVOLUTION!
ALL THE SOFTWARE YOU NEED!
GETTING STARTED WITH
LINUX
• MORE POWERFUL • MORE SECURE • MORE FU
LEARN HOW TO SET UP A LINUX SYSTEM TO:
• Listen to Music • Play Games • Process Photos
• Surf the Web • and Much More!
START
2020 Edition
LINUX MAGAZINE Special
WWW.LINUX-MAGAZINE.COM
301 BEST BASH COMMANDS
LINUX SHELL
HANDBOOK
2021 Edition
SUPERCHARGE
YOUR LINUX SKILLS
Power at Your Fingertips
• Pipe and redirect output
• Monitor processes
• Create custom scripts
Keep this guide as a permanent reference!
LINUX NEW MEDIA
WWW.LINUX-MAGAZINE.COM
FREE DVD!
LibreOffice Full Version
Become a LibreOffice Expert!
2020 Edition
LibreOffice
Dive deep into the world's greatest free office suite
Write Your Own LO Macros
Save time and automate common tasks
Digital Signatures
Lock down your private documents
Edit and Save MS Office Files
Create Professional:
Text Documents
Spreadsheets
Presentations
Databases
Replace MS Office an
Google Docs!
LibreOffice
Includes full versions fo
Windows, macOS, and Li
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 MAGAZINE Special
WWW.LINUX-MAGAZINE.COM

MakerSpace

An introduction to electronic weighing

Weighing In

Create your own weighing device with easily available components and open source software. By Andrew Malcolm

In this article, I discuss the design of a compact and portable workshop balance for various single-load-cell weighing applications constructed with a small load cell, an instrumentation amplifier, an excitation supply, a microcontroller, a display, and a serial port for debugging. Throughout, I used Linux and open source software, and I provide code samples, with directions for finding the complete code online.

History

Ancient civilizations used simple balances to compare weights for trading in precious metals, spices, salt, and the like. Today's civilization is no less dependent on knowing the weight of objects. It is hard to imagine a day in which the knowledge of weight does not take part: from the morning visit to the bathroom scales, a trip to the supermarket, baking a cake, to weighing baggage at the airport – the list of times weight plays a part in our lives seems endless. Today, we've moved away from mechanical balances, for the most part, obviating the need for ready reference weights.

Today's weighing equipment is usually based on electronic signals from strain gauges. These sensors are thin-film resistors whose resistance varies in response to tension or compression. When bonded to a mechanical structure subject to the force of an applied mass,

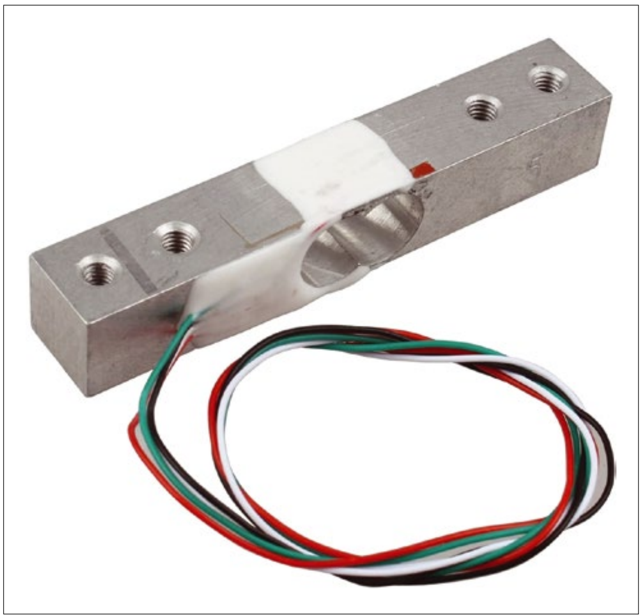


Figure 1: A typical modern load cell. The strain gauges are covered with a white flexible coating to protect them.

Lead Image © Carlos Velayos, 123RF.com

Author

Andrew Malcolm (MIET, CEng) works as a lead hardware and firmware engineer for Guru Systems (*https://www.gurusystems.com/*), a fast-growing IoT hardware and SaaS company working on low-carbon-energy projects. In his spare time, he likes to combine software engineering with his first love, hardware engineering. With all the open source tools available, he is never short of things to design. The Raspberry Pi has proved to be a source of inspiration, and to date, Andrew has designed five different add-ons, or HATs. He is currently working on microstepping motor drives for a Pi-based laser cutting machine. You can contact him at *andrewrussellmalcolm@gmail.com*.

the resistance of a strain gauge will change proportionally in response. Practical weighing systems use more than one strain gauge, and these are generally bonded to a metallic billet in a controlled manner to form a more complex electrical circuit designed to eliminate nonlinearities and temperature effects. These billets are known as load cells and are available commercially with working ranges from a few grams to hundreds of tonnes.

Load cells (Figure 1) are built into the structures of complete weighing systems such as scales, weighbridges, hoists, and counting scales in such a way that they can be replaced easily should the need arise. Although load cells are transducers with no moving parts, of course they do deflect slightly under load, and that allows mechanical endstops to be used to limit the deflection and protect the cell from overload. It should be noted that larger weighing machines such as weighbridges will employ more than one load cell, a typical arrangement being a rectangular platform with a load cell at each corner. The signals from each load cell are summed electronically to provide a total weight. Often, a range of load cells of different capacities will share the same basic billet, machined in different ways.

Load Cell Operation

In most load cells, four strain gauges are employed – two in tension and two in compression – in a symmetrical manner to eliminate temperature effects (Figure 2). The strain gauges are arranged into a Wheatstone bridge (Figure 3).

A strain gauge is a sensor whose resistance varies with applied force, so that if you can measure resistance, you can measure that force. The conducting material (often constantan) is deposited onto an insulating substrate such as polyimide. The conductor is designed in a serpentine shape (Figure 4) such that extension by applied force in the longitudinal direction is much higher than that in the lateral direction. This arrangement reduces sensitivity to off-axis load. Pads are provided at the ends of the conductor so that lead-out wires may be soldered or otherwise bonded to them.

The Wheatstone bridge requires a voltage across it to provide an output, and that voltage is called the excitation voltage. The output of the bridge is a differential, which slightly complicates the electronics, as you will see later. At zero load, the difference between the two outputs should be zero, and as load is applied, one arm will go positive while the other goes negative. The differential output at full scale is measured in millivolts per volt (mV/V) (i.e., a ratio of output versus the excitation voltage). Excitation voltages are generally in the range of 5V to 20V, and a typical load cell might have an output of 2mV/V, making 20mV output at full scale for 10V excitation. From this, it can be seen that you are dealing with very small voltages, especially when you require high precision. The upside of this arrangement is that much the same electronics can be applied to a wide variety of different weighing applications, because the load cell's maximum output will be similar in all cases.

The nominal resistance of the strain gauges is on the order of a few hundred ohms, so a bridge made up of four 400-ohm gauges will have a resistance (R) of 400 ohms, as seen by the excitation supply; thus, it can be see that some power is required to excite the load cell, which can be a challenge for battery-operated systems. In this case, lowering the excitation

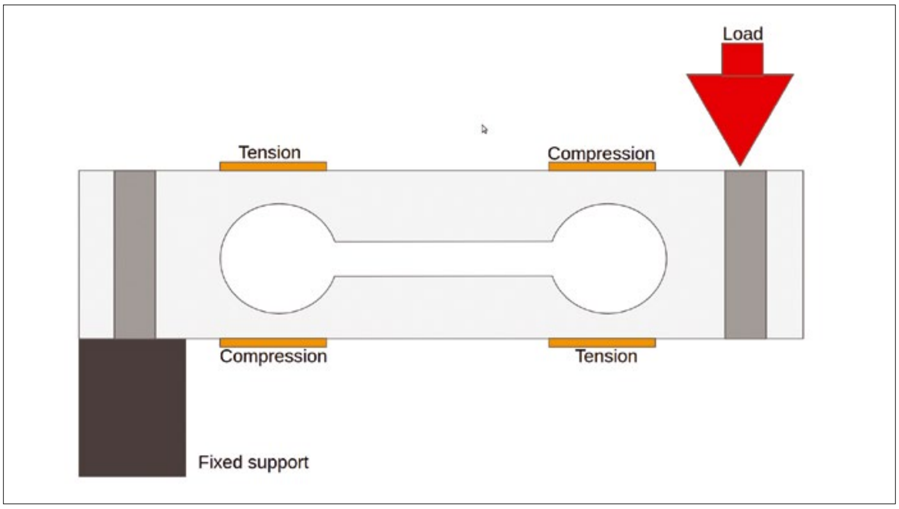


Figure 2: The arrangement of the strain gauges on the load cell billet and the forces that act on them. The "dumbbell" shape cut out of the billet makes the metal thinner in the vicinity of the strain gauges, increasing sensitivity.

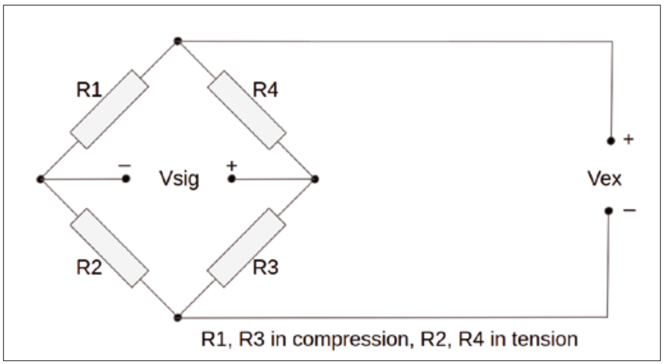


Figure 3: This electrical arrangement is attributed to Charles Wheatstone [1], a Victorian physicist. It is very useful in this type of measurement because it allows the user to "null out" the standing voltages in a system and thus make accurate measurements of small changes.

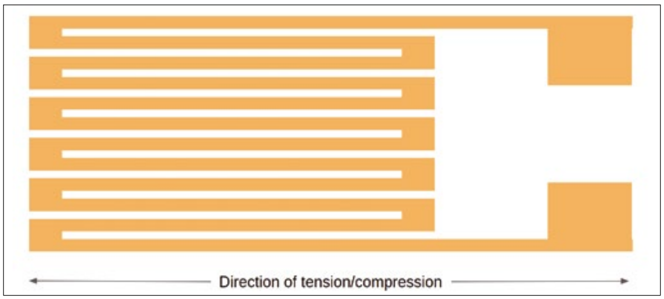


Figure 4: Strain gauge conducting material design.

voltage saves power at the expense of full-scale output and, ultimately, precision. Although increasing the excitation voltage might seem attractive from the point of view of increasing load cell output, there comes a point when the strain gauges will self-heat, causing local expansion of the load cell billet and the attendant inaccuracy.

In a little more detail, the output of the load cell can be calculated in the following manner: V- = R2/(R1 + R2) x Vex, whereas V + = R3/(R3 + R4) x Vex, so the differential output voltage is:

$$V_{sig}=[R3/(R3+R4)-R2/(R1+R2)]\times V_{ex}$$

Tare and Span: Calibration

One factor that must be considered when choosing a load cell for an application is tare, which refers to the weight of any structure supported by the load cell before any object to be weighed is placed on it. This tare might be the mixing bowl on a kitchen scale before ingredients are added or the platform required to support a multitonne truck on a weighbridge. Tare can represent a large portion of the ultimate full-scale range of a system and must be taken into account at the design stage.

Further removal of tare is sometimes required during normal operation: Again, using the example of a weighbridge on which a truck is to be loaded with goods, the initial zero point would be with the weighbridge empty. Once an empty truck is driven onto the weighbridge, it is convenient to again "zero" the weight display so that as goods are added to the truck, the display reflects what has been added, without the complication of the truck's tare. In the case of a weighing system designed to dispense ingredients in a fixed ratio (batch weighing) into a vessel (e.g., flour, water, yeast, and sugar in a bakery), the tare may be reset several times as the batch is assembled with the different ingredients.

"Span" is the term used to describe the full range of a weighing machine. The desired span is application dependent, and the load cell(s) must be chosen to have sufficient capacity for the required span plus tare, as described above. The stated output of the load cell at full scale is generally a nominal figure only, meaning the instrument must be calibrated before use at intervals during its operational life because of various mechanical and temperature effects. For a small instrument, this is a simple undertaking, reference weights being readily available at varying levels of precision from commercial sources [2], some of which can be traced back to national and international standards bodies.

For larger systems, calibration can be a complex and expensive exercise because you have no way to avoid the use of dead weights for calibration, and moving such weights around is time consuming and expensive.

Precision

Precision of a weigh scale is often quoted in "counts." For example, in a machine designed to weigh up to 10kg at 10,000 count precision, the least significant digit would represent 1g. In reality, precision at this level is difficult to achieve and often not required. If the load cell in such a machine has an output of 10mV at full scale, one count is represented by 1µV. Such low-level signal changes can be difficult to detect, especially in the presence of electrical noise and, indeed, vibration from the surrounding environment, which can render the weight signal unstable to begin with. Long-term stability of the load cell itself (known as creep), temperature effects, and tiny voltages generated between dissimilar metals in the wiring between the load cell and the amplifier can all contribute to errors in measurement and mandate regular calibration of the machine, especially if it is used at a point of sale. In these cases, various regulatory bodies [3] enforce calibration intervals to ensure customers get what they pay for.

Workshop Weighing System

As an example design, I wanted to produce something compact and portable that might prove useful in my workshop. I found a small load cell, freely available on eBay.com or other online sources, with a capacity of 100g (3.5oz) at low cost. The load cell is physically quite small (5x1x0.6cm) and easy to mount, and at such low capacity, it is easy to assemble into a plastic enclosure to produce a practical instrument. The load cell is one of a family of force transducers, so it would be easy to adapt the design to weigh heavier items for different applications.

The electronic design comprises four basic elements: an instrumentation amplifier, an excitation supply, a microcontroller, and a display.

Analog Front End

A number of application-specific integrated circuits are available for weigh scale applications that comprise a differential instrumentation amplifier and a 24-bit sigma-delta analog-to-digital converter (ADC). Some of these integrate other functions, such as an excitation regulator, but in essence, they perform the same task.

One of the most popular of these ADCs for weigh scales is the HX711 [4] from Avia Semiconductor, a Chinese company. It is readily available from Aliexpress.com and eBay.com, both as a bare chip and mounted on a simple printed circuit board (PCB). I assume this chip forms the basis of many of the kitchen scales and the like manufactured in China. I did some initial prototyping with this chip, and it seems very capable. Certainly, if you want to hook up a load cell to an Arduino or a Raspberry Pi, the datasheet provides example drivers in C and assembler, and you can get something up and running in a few minutes.

An alternative part from a more established manufacturer is the ADS1232 [5] from Texas Instruments. This part is available through more traditional distribution, such as mouser.com and digikey.com. The datasheet (and associated application notes) is somewhat more detailed, with an application circuit provided for a weigh scale. The ADS1232 lacks the excitation regulator of the HX711, but that in itself has some limitations, and a separate excitation regulator is not difficult to implement and is more flexible and potentially more stable. Interestingly, both of these chips have a very similar digital interface, suggesting they share a common heritage.

My choice for this design was the somewhat more expensive ADS1232 with a separate excitation regulator implemented with a REF5040 [6] (also from Texas Instruments), a high-stability reference supply chip capable of supplying 4V at 10mA directly to the load cell.

Display

For the display, I wanted a large, bright, clear output that was cost effective. I also wanted something that was easy to drive and required few pins from the microcontroller. The display I chose was a six-digit LCD with a white LED backlight from Hobby Components [7] that requires only three lines to drive. The digits are about 1.5cm (0.6in) in height, with a battery state indicator to the right of the display. It is based on the HT1621 chip by Holtek [8], and the datasheet is available from their website.

Microcontroller

Once all the other parts of the design were chosen, attention could turn to selecting a suitable microcontroller. Because I'm already familiar with ST Microelectronics' range of ARM-based controllers, this was an obvious place to start. However, I also wanted a low-cost solution that was scaled appropriately for this application. I was pleasantly surprised to find that the ST's "Value Line" controllers include a device that runs at 48MHz without an external crystal, has 20 pins (plenty for this application), and 16KB of flash program memory. Programming and debugging use ST's two-wire ST-Link interface, and a Linux-based IDE, the STM32CubeIDE, integrates well with ST-Link. This device is available for less than a dollar. USB-based ST-Link programmers are available online for just a few dollars, as well. The exact device is the STM32F030x4 [9].

Design

Schematic capture and PCB layout were both performed by KiCad, a free and open source CAD tool originally developed at CERN [10]. It really is an excellent suite of tools and handles the entire process of electronic design from schematic capture (Figure 5) right through to generating files for manufacture. It even has a 3D viewer that generates a panable/rotatable image of your design, including the components. Although PCB assemblies are in some ways two dimensional, the use of the 3D viewer has saved me from mechanical clashes that are not apparent from the two-dimensional design perspective. You can export the 3D model as a STEP file and import that into 3D CAD tools such as FreeCAD to build up more complex assemblies (e.g., aiding the design of parts suitable for 3D printing, such as enclosures).

Many excellent PCB companies online will build good-quality PCBs in a few days for less than $5 (EUR6, £5), so building prototypes or experimental PCBs is not prohibitively expensive.

A four-pin terminal block is provided for connecting the load cell: two excitation pins and two signal pins. The excitation is provided by the REF5040, and the signal pins go to the ADS1232 through a simple low-pass filter. The ADS1232 needs a handful of passive components for filtering and stability, and its digital interface is three lines that go directly to the microcontroller. The microcontroller in turn drives the display with its three-line interface, with enough spare I/O pins on the controller to provide additional facilities (i.e., a battery monitoring circuit, a temperature input from the REF5040, a serial port for debugging and/or data logging, and two buttons used for tare and span setting). A pin switches or dims the display backlight.

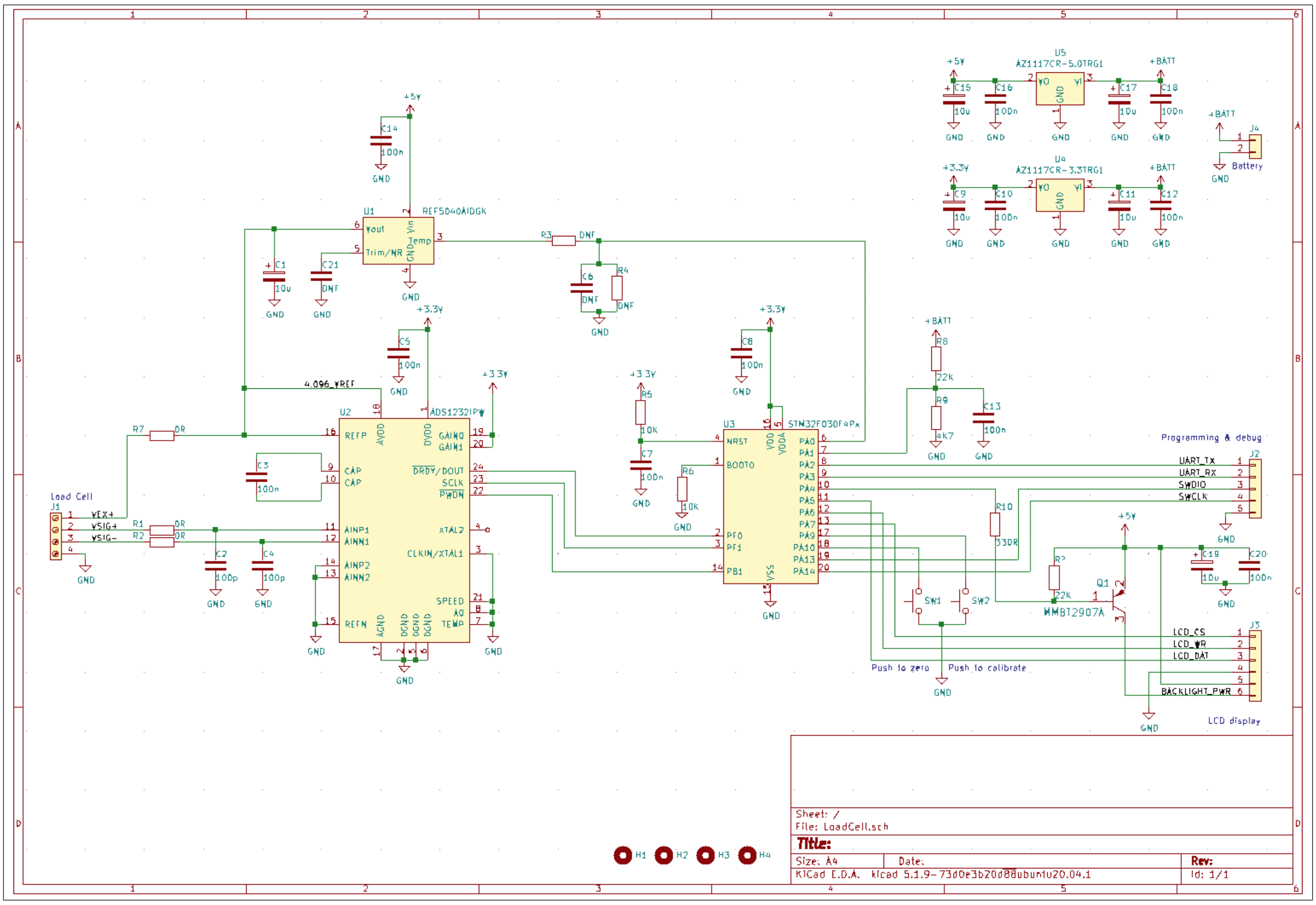


Figure 5: The full schematic for the final design.

Power is supplied by a PP3 (9V) battery, and two linear regulators provide 3.3V for the digital electronics and 5V for the analog section, giving some isolation between the two to minimize the effects of digital electrical noise. The 5V analog supply allows the REF5040 to generate a stable 4.1V for the load cell excitation. The 9V supply greatly simplifies the power supply design, allowing the use of linear regulators. If a lower supply were used (e.g., two lithium ion cells), some form of switching regulator would be required to boost the voltage to a level suitable for the load cell excitation, with all the attendant noise problems that could bring.

The PCB layout itself is fairly straightforward. Having decided the display would mount directly above the PCB, I saw that it was fairly apparent that most components could go under the display, with only switches and connectors requiring a margin around the edge. Therefore, I arrived at a size of 100x50mm (4x2 inches), with a two-layer PCB being more than adequate for such a simple circuit. Close attention was paid to layout of the analog section, care being taken to distance it from digital lines where possible, placing decoupling capacitors close to the analog chips and creating generous ground planes on the top and bottom layers. A 3D rendering of the resulting PCB is shown in Figure 6.

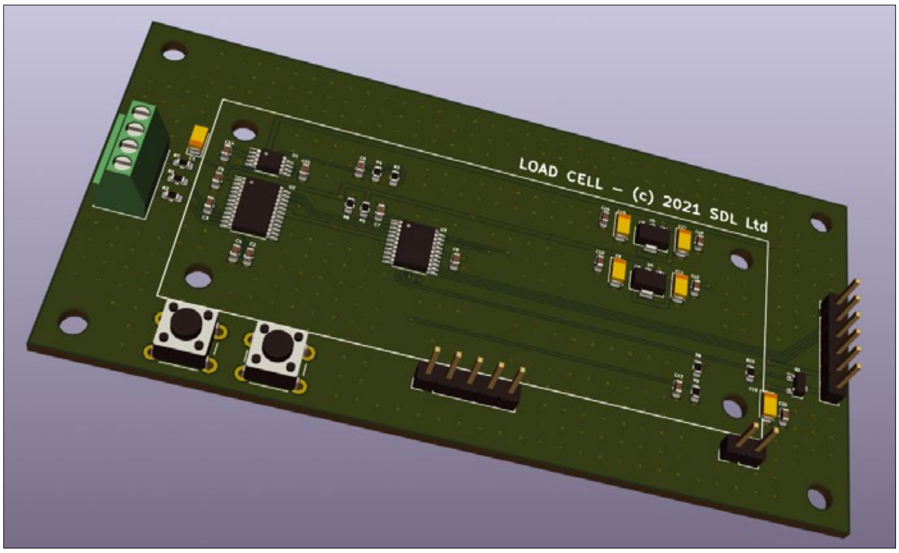


Figure 6: A 3D view of the load cell PCB.

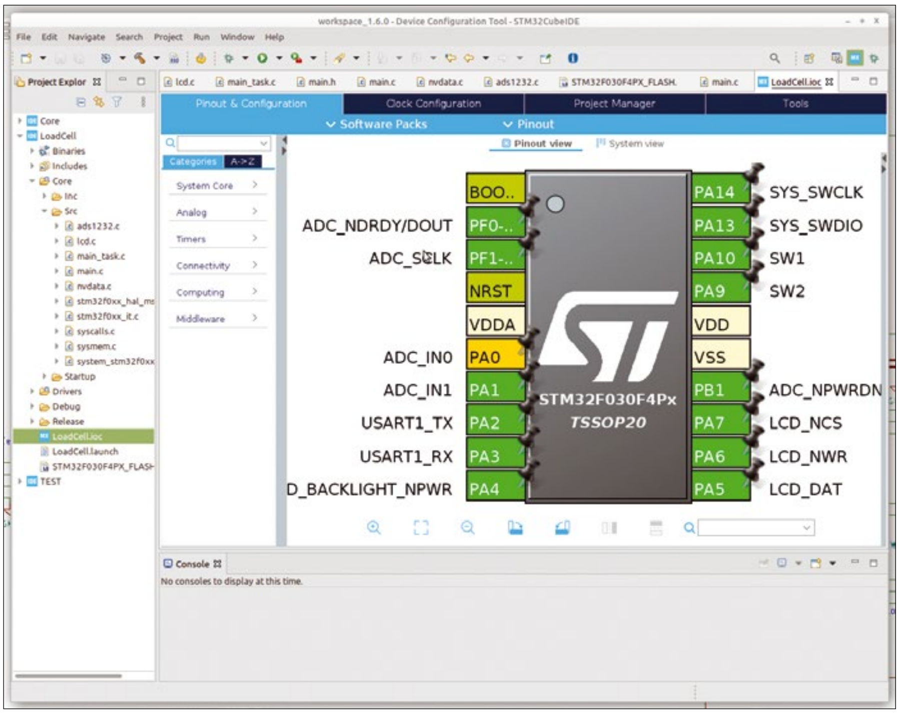


Figure 7: The project pin configuration in STM32CubeIDE.

Software Development

The use of IDEs can be controversial and very much a matter of taste, and it's certainly possible to do this type of microcontroller development without one. The ARM compilers and standard libraries can be downloaded from your distro's repository, and you're off, using any editor that suits you and Make or Cmake – again, your choice. Once you have a compiled binary, utilities for ST-Link allow you to program your device, and you can use GDB (the GNU debugger) to debug you program.

That said, ST's STM32CubeIDE, based on Eclipse, does streamline the process by integrating ST's CubeMX tool, a utility that allows you to configure your microcontroller and generate a software framework that does all the initialization and leaves you with a blank `main()` function to add to your code. You can label the pins of the microcontroller (bonus points if you use the same names as on the schematic!).

The pin configuration for this design is shown in Figure 7. Once saved, the IDE generates a set of `#define` directives for the I/O pins that you can use in your code and a complete set of initialization routines. At this point, you can continue to use the IDE or ignore it and use Make with the generated makefile. However, if you stay with the IDE and have your hardware connected by an ST-Link programmer, a single mouse click compiles, downloads, and runs your code. This level of preconfiguration – including, if you want, the inclusion of a real-time operating system (RTOS) such as FreeRTOS – can leave you free to concentrate on your application code. In a commercial environment, time to market is everything, and time savings like this can be invaluable.

Serial I/O and printf

My hardware design and microcontroller configuration includes a serial port for debugging. Being able to output data and wait for keystrokes during debugging can be invaluable as an alternative to, or as an adjunct to, a debugger. The STM32 libraries make it easy to override the low-level routines that `printf` and

Listing 1: Overriding Low-Level I/O Primitives

```
/** read from the serial port */
int _read(int file, char *ptr, int len)
{
  HAL_UART_Receive(&huart1, (uint8_t *)ptr, 1, -1);
  return 1;
}

/** write to the serial port */
int _write(int file, char *ptr, int len)
{
  int DataIdx;

  for (DataIdx = 0; DataIdx < len; DataIdx++)
  {
      if (*ptr == '\n')
      {
          HAL_UART_Transmit
            (&huart1, (uint8_t *)"\r\n", 2, -1);
      }
      else
      {
          HAL_UART_Transmit
            (&huart1, (uint8_t *)ptr, 1, -1);
      }
    ptr++;
  }
  return len;
}

/** wait for a keystroke */
uint8_t kbhit(void)
{
    return __HAL_UART_GET_FLAG(&huart1, UART_FLAG_RXNE);
}
```

Listing 2: Reading the ADC

```
/** wait for ADC sample ready and read it out */
uint32_t ADS1232_Read(void)
{
  int32_t value = 0;

  // wait for DOUT to go low : data ready
  while (HAL_GPIO_ReadPin(ADC_NDRDY_DOUT_GPIO_Port, ADC_NDRDY_DOUT_Pin))
    ;

  for (uint8_t i = 0; i < 24; i++)
  {
    value = value << 1;

    // send clock pulse
    HAL_GPIO_WritePin(ADC_SCLK_GPIO_Port, ADC_SCLK_Pin, 1);
    HAL_GPIO_WritePin(ADC_SCLK_GPIO_Port, ADC_SCLK_Pin, 0);

    // read the data in
    if (HAL_GPIO_ReadPin(ADC_NDRDY_DOUT_GPIO_Port, ADC_NDRDY_DOUT_Pin))
    {
      value |= 1;
    }
  }

  // reset DOUT
  HAL_GPIO_WritePin(ADC_SCLK_GPIO_Port, ADC_SCLK_Pin, 1);
  HAL_GPIO_WritePin(ADC_SCLK_GPIO_Port, ADC_SCLK_Pin, 0);

  // 24-32 bit 2s complement conversion
  value <<= 8;
  value /= 256;

  return (value);
}
```

`scanf` eventually call and redirect them to a serial port (Listing 1). Once this is done, these functions are available from whatever terminal (e.g., the Linux `minicom` tool) you choose to connect to the serial port. This serial port could also be used in an application for batch weighing, in which an external computer or programmable logic controller (PLC) monitors the weight signal on a continuous basis.

ADC Driver

The ADS1232 has a very simple two-wire serial interface with no configuration registers, so all setup (clock speed, gain, etc.) happens from pins on the chip. To read the data from the chip, the software must first wait for a valid sample, which it does by waiting for the DOUT (data out) line to go low (Listing 2). At this point, data is clocked out in a serial manner by pulsing the clock pin from low to high and reading the resulting data bit from DOUT. This process is repeated 24 times to clock out a complete 24-bit sample. A further clock pulse resets DOUT to its “data ready” function.

Filtering Data

Before displaying the data, some sort of filtering is usually required that removes fluctuations in the weight display caused by noise and vibration, but filtering comes at the cost of response time. The raw samples from the ADC take 100ms to acquire, so any form of low-pass filtering will slow

Listing 3: Averaging ADC Readings

```
#define SAMPLES 10

int32_t samples[SAMPLES];
int32_t average = 0;
uint8_t sample = 0;

// main acquisition loop
while (true)
{
  samples[sample++] = ADS1232_Read();
  if (sample == SAMPLES)
  {
    sample = 0;
  }

  average = 0;
  for (uint8_t i = 0; i < SAMPLES; i++)
  {
    average += samples[i];
  }

  average /= SAMPLES;

  // pass the averaged data to the display here
}
```

down the response time into the region by fractions of a second. In the case of a workshop balance, a couple of seconds for the reading to settle is not a problem, but in other cases (e.g., a batch-weighing application), every second adds to production time and therefore costs money. Filtering and ultimate accuracy will always be a tradeoff. In this application, 10 samples are simply averaged before the result is passed for display (Listing 3).

Practical Precision

In terms of practical precision, you must keep a few things in mind. Of the theoretical 24 bits of precision available from the ADC, the datasheet itself admits that only 19 bits are useful at the highest gain setting. The load cell I have chosen has a maximum output of 4mV (1mV/V, with an excitation of 4V), so it uses only one fifth of the available range (20mV) of the ADC. That's about 100,000 counts. Practically, I have found I can achieve stable readings to 10,000 counts. For a 100g scale, that means one count is equivalent to 10mg. The current software limits that further, giving a resolution of 100mg (i.e., 1,000 counts), which seems more than adequate for the requirements of a general-purpose laboratory scale. There is certainly scope for better filtering and other techniques to improve the precision, but at some point the limitations of the (cheap) load cell will become the limiting factor.

Display Driver

The display has a three-wire serial interface. The chip select (CS) line must be asserted before data can be written, and once that is done, data is presented on the data (DAT) line, being clocked into the chip by asserting the write (WR) line and then negating it (Listing 4). The interface is too slow to keep up with the microcontroller at full speed, so busy-wait loops are necessary to slow the WR line pulses down to just over 3µs.

The display chip has a small amount of internal memory, and the individual bits in this memory correspond to the different segments in the display characters, as well as decimal points and the battery state indicator. Each transaction with the chip is called a command, and the first three bits of that command determine the type of command. The rest of the transaction comprises an address in the internal memory and the data to be written.

The process of writing a complete digit involves conversion from the required ASCII digit to the memory bit pattern to illuminate the display correctly. This code is not complex, but quite long, so I won't include it here. I refer the interested reader to the source code on my GitHub page [11].

Storing Calibration Parameters

The design provides no non-volatile storage for persistent data such as calibration parameters. In a larger system,

Listing 4: Writing a Single Bit to the Display

```
/** */
static void LCD_WriteBit(bool bit)
{
  // set the data line
  HAL_GPIO_WritePin(LCD_DAT_GPIO_Port, LCD_DAT_Pin, bit);

  // assert the WR line
  HAL_GPIO_WritePin(LCD_NWR_GPIO_Port, LCD_NWR_Pin, 0);

  // busy wait
  for (uint8_t i = 0; i < 50; i++)
    ;

  // negate the WR line
  HAL_GPIO_WritePin(LCD_NWR_GPIO_Port, LCD_NWR_Pin, 1);

  // busy wait
  for (uint8_t i = 0; i < 50; i++)
    ;
}
```

Listing 5: Nonvolatile Storage Code

```
/*
 * nvdata.c
 *
 *  Created on: 19 Mar 2021
 *      Author: andrew
 */

#include "main.h"

extern uint32_t _nvdata[1024];
 // start of non-volatile data (last block of main flash)
 defined in linker file

/** */
void NVDATA_SaveCalibrationParams
 (uint32_t span, uint32_t zero)
{
  HAL_FLASH_Unlock();

  FLASH_EraseInitTypeDef eraseInit;

  eraseInit.TypeErase = FLASH_TYPEERASE_PAGES;
  eraseInit.NbPages = 1;
  eraseInit.PageAddress = (uint32_t)&_nvdata[0];

  uint32_t sectorError;
  printf("erasing flash page at %p\n", &_nvdata[0]);
  HAL_StatusTypeDef res = HAL_FLASHEx_Erase
   (&eraseInit, &sectorError);

  if (res != HAL_OK)
  {
    printf("FLASH erase error %d\n", res);
  }

  res = HAL_FLASH_Program
   (FLASH_TYPEPROGRAM_WORD, (uint32_t)&_nvdata[0], span);

  if (res != HAL_OK)
  {
    printf("FLASH program error %d\n", res);
  }

  res = HAL_FLASH_Program
   (FLASH_TYPEPROGRAM_WORD, (uint32_t)&_nvdata[1], zero);

  if (res != HAL_OK)
  {
    printf("FLASH program error %d\n", res);
  }

  HAL_FLASH_Lock();
}

/** */
void NVDATA_LoadCalibrationParams
 (uint32_t *span, uint32_t *zero)
{
  *span = _nvdata[0];
  *zero = _nvdata[1];
}
```

it is customary to have an external SPI flash memory or EEPROM chip to provide this function. The current design uses a small portion (one page, 1KB) of the microcontroller's program flash memory as non-volatile storage (Listing 5), and you can tell the linker to exclude a portion of this flash memory from program memory (usually a page at the highest address available). That block can then be labeled in the linker file so that it is accessible from program code, and primitives in the STM32 libraries can erase pages and read/write bytes or words from that area. This technique saves the expense of an external memory chip and the I/O pins that would be required to drive it. Listing 6 from the linker file shows how to reserve a portion of flash memory for parameter storage.

Calibration

The design has two push buttons for calibration: one for tare and one for span. In the main acquisition loop (Listing 7), the code looks for push-button events. If the tare button is pushed, the current reading is stored and subtracted from subsequent readings to provide a

Listing 6: Linker File Extract

```
/* Entry Point */
ENTRY(Reset_Handler)

/* Highest address of the user mode stack */
_estack = ORIGIN(RAM) + LENGTH(RAM);   /* end of "RAM" Ram type memory */
_nvdata = ORIGIN(NVDATA);

_Min_Heap_Size = 0x200 ;  /* required amount of heap  */
_Min_Stack_Size = 0x400 ; /* required amount of stack */

/* Memories definition */
MEMORY
{
  RAM    (xrw)    : ORIGIN = 0x20000000,  LENGTH = 4K
  FLASH  (rx)     : ORIGIN = 0x8000000,   LENGTH = 15K //reduce flash by 1K
  NVDATA (rx)     : ORIGIN = 0x8003c00,   LENGTH = 1K //define the nv region
}

/* Sections */
SECTIONS
{
 ### ommited ###

  .nvdata : // give the nv region a name so it can be referenced from code
  {
    KEEP(*(.nvdata))
  } >NVDATA

 ### ommited ###
}
```

zero output. Similarly, if the span button is pushed, the required span factor is calculated and stored (in this case, the code assumes a 100g calibration weight has been placed on the scale).

Completed Unit

I used FreeCAD to design an enclosure for the completed instrument and had it printed by a local 3D printing house. FreeCAD is another open source package, and every time I return to FreeCAD, I find the developers have made another step increase in functionality. It really is an exemplary open source project. The STEP files created were sent directly to a local 3D print house and were printed in just a few days.

I purchased a suitable stainless steel pan with a tare weight of just 50g, and the completed assembly can be see in Figure 8. I've left off the front panel to show the mounting arrangement of the PCB and the display. The load cell is mounted behind, with a vertical riser coming up through an aperture in the enclosure. A small circular plastic plate is attached to that, on which the pan sits.

Wrap-Up

I've presented here some background on the theory and application of industrial weighing. I've shown a practical design for a workshop balance that is flexible enough to be applied to many single-load-cell weighing applications and, in the process, shown that Linux and open source software make an excellent choice for engineering applications. I hope this example encourages you to experiment with strain gauges and load cells and come up with your own innovative applications. The complete design is available on my GitHub page, and I can be contacted by email (see my bio).

Finally, I'd like to dedicate this article to my father, who started and ran his own industrial weighing company for many years and taught me much of what I know. ■■■

Listing 7: Calibration

```
int32_t zero = 0;
int32_t span = 1;

// main acquisition loop
while (true)
{
  // get averaged data here

  if (sw1PushedEvent())
  {
    zero = average;
    NVDATA_SaveCalibrationParams(span, zero);
  }

  if (sw2PushedEvent())
  {
    raw = (average - zero) * 100;

    span = raw / 1000;

    NVDATA_SaveCalibrationParams(span, zero);
  }

  display = ((average * 100 - zero * 100) / span);

  LCD_PrintNumber(display, 0);
}
```

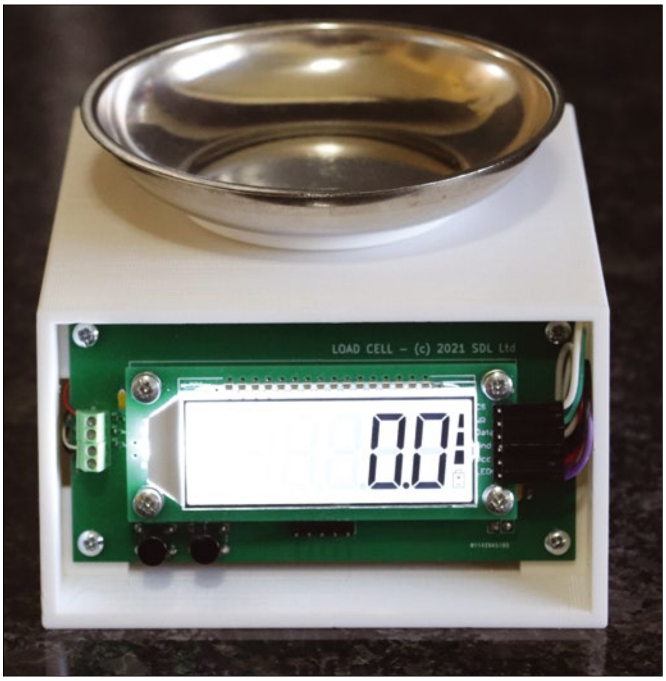


Figure 8: The 3D-printed enclosure with the front panel removed.

Info

[1] Charles Wheatstone: *https://en.wikipedia.org/wiki/Charles_Wheatstone*

[2] OIML and ASTM weights: *https://static.fishersci.com/cmsassets/downloads/segment/Scientific/pdf/MettlerToledo/11796035_OIML_weights.pdf*

[3] International Organization of Legal Metrology: *https://www.oiml.org/en*

[4] HX711: *https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf*

[5] ADS1232: *https://www.ti.com/lit/ds/symlink/ads1232.pdf*

[6] REF5040: *https://www.ti.com/lit/ds/symlink/ref5040.pdf*

[7] HT1621: *https://hobbycomponents.com/displays/1053-ht1621-6-digit-7-segmenr-lcd-module-white-backlight*

[8] HT1621 datasheet: *https://datasheet.lcsc.com/szlcsc/Holtek-Semicon-HT1621B_C7873.pdf*

[9] STM32F030x4: *https://www.st.com/resource/en/datasheet/dm00088500.pdf*

[10] KiCad: *https://home.cern/news/news/computing/kicad-software-gets-cern-treatment*

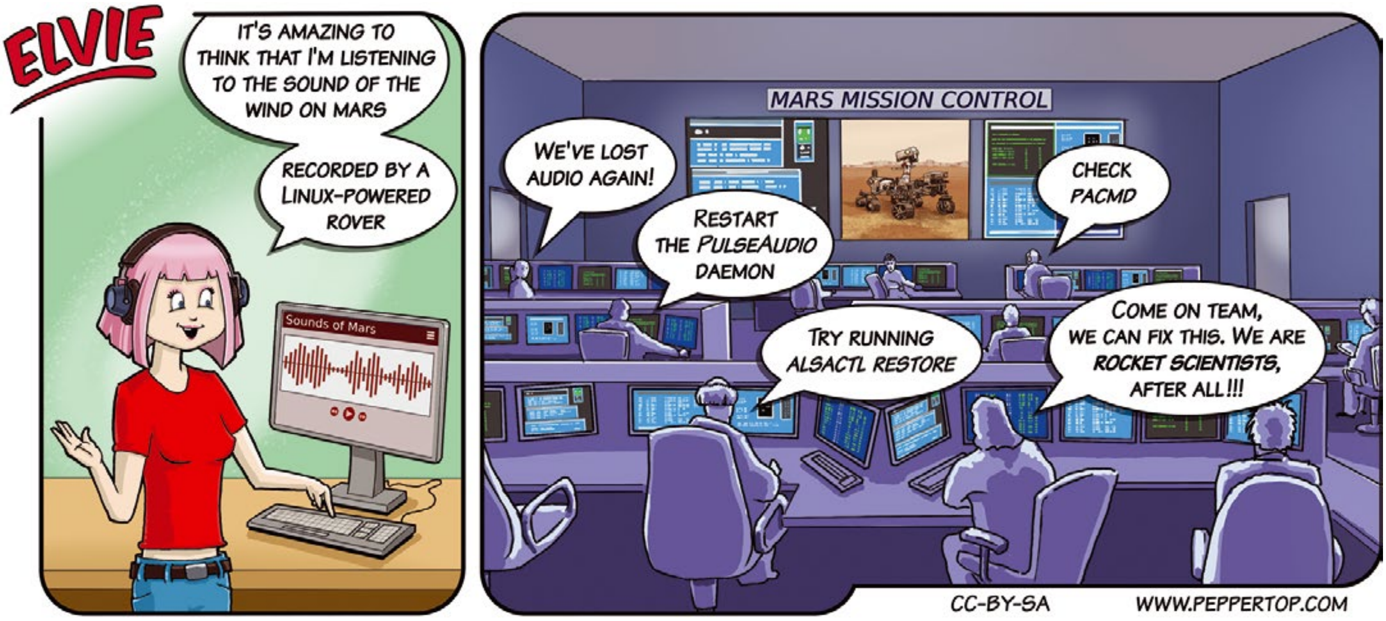
[11] Author's GitHub project page: *https://github.com/andrewrussellmalcolm/loadcell*

■■■

The privacy issues surrounding search engines are well known to Linux users. Do one search and the ads follow you everywhere. The open source community recoils at such displays, but where to turn? Maybe you have heard of DuckDuckGo, but have you heard of Ecosia, the search engine that uses a portion of their profits to plant trees, or Searx, a search engine that searches the open APIs of 70 other search engines? This month we introduce you to five alternative search tools, and we even show you how to set up an engine that searches your own website. Elsewhere in Linux Voice: Document your desktop with the Menyoki screencast tool, and organize your office workflow with SeedDMS.

Image © Olexandr Moroz, 123RF.com

LINUXVOICE


ELVIE
IT'S AMAZING TO THINK THAT I'M LISTENING TO THE SOUND OF THE WIND ON MARS
RECORDED BY A LINUX-POWERED ROVER
Sounds of Mars
MARS MISSION CONTROL
WE'VE LOST AUDIO AGAIN!
RESTART THE PULSEAUDIO DAEMON
CHECK PACMD
TRY RUNNING ALSACTL RESTORE
COME ON TEAM, WE CAN FIX THIS. WE ARE ROCKET SCIENTISTS, AFTER ALL!!!
CC-BY-SA
WWW.PEPPERTOP.COM

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®.

MADDOG'S DOGHOUSE

This month, maddog charts the changing norms in dress code over his lengthy career in the tech industry. BY JON "MADDOG" HALL

Dress for Success

My favorite apparel usually consists of some type of printed T-shirt (not white, please!) and a pair of shorts. This is what I normally wear between the dates of April 1 and November 30. Snow has often been falling before I put on a pair of long pants and a long-sleeved shirt.

Of course, I also dress up for formal occasions, such as when I perform a wedding ceremony or funeral (the latter have, fortunately, been rare), but I have not owned a suit that fits for the past two decades.

This has not always been true. When I was a cooperative education student at Drexel University (the Drexel Co-op provides students with professional employment experiences as part of their degree), I wore a white shirt (with pocket protector), tie, and long pants to work. This was not just because I worked at the Western Electric Company, a very conservative organization, but also because the white shirt, tie, and pocket protector told the people on the manufacturing floor that you were either part of management or an engineer, and both roles held a modicum of respect and power. People in the shop were supposed to listen to you.

It was while I was a cooperative education student that I decided I would never shave again. I had shaved my beard in order to get the job (as I said, Western Electric was a very conservative company in the days of the Vietnam War, hippies, and long hair), but after about two months, my boss's boss (a retired Air Force colonel) told me to grow it back because he already had "two teenage sons." I never shaved again.

I also had an experience looking for that first postgraduate job. I interviewed with a company that stated in their policies that their employees would have no facial hair or hair touching their collars, and that they would wear a brown or black suit, white shirt (no stripes), narrow dark tie, neatly clipped fingernails … it went on and on. After I read about this dress code, I knew I would not fit in, but I went to the interview anyway to ask why they were so strict.

The answer was simple. This company would do *all* of the data processing for companies run by 55- to 65-year-old CEOs (write the programs, buy the computers, run the computers, etc.), and all the CEOs would have to do is write the check. These CEOs wanted someone that looked like them, not "a bead-and-sandal-wearing hippie."

That company, Electronic Data Systems (EDS), was a fledgling computer service company run by H. Ross Perot, a very conservative person who eventually ran for president of the United States (and lost).

I had many different jobs with different levels of dress codes over the years. For technical people, these dress codes were often relaxed.

Eventually I worked for Digital Equipment Corporation (DEC). If you were an engineer, the dress code was fairly relaxed, but if you were a product manager, you were expected to "dress up" because sometimes you might be hosting customers at the facility.

When we started the Unix group, we were a small group of engineers housed in a set of buildings that were usually used by marketing groups. One day a young engineer who normally brought his lunch or ate snacks out of machines decided to go to the cafeteria. A Dead Head, he was in shorts, a tie-dye T-shirt, and barefoot. He ran upstairs, filled up a tray with food, and ran down to his office to eat it. He did not notice the stunned silence of the marketing people in the cafeteria, all dressed in suits and ties.

The next day there was a sign in the cafeteria: "Shirts and Shoes Required."

DEC sent me out to Palo Alto, Calif., as a product manager. One time, I was told that EDS was coming for a meeting, and I had to wear a suit and tie because "*It was EDS!*" It so happened that I had a suit and tie that fit, so this was not a big deal. That day I came into work ready to do my presentation. I even took out my earrings so I would not offend the customer.

Before I entered the conference room, I looked in a small window and saw all of the EDS people dressed in shorts, T-shirts, and sandals. My associate had taken off his coat and tie, and on the whiteboard there was a big picture of a tie in a red circle with a line through it – no ties.

I took off my coat and tie, put my earrings back in, and went in to present.

EDS' policy was really "dress as the customer dresses." Since the EDS people were in California, they dressed as they thought California tech people dressed, and so did we. ■■■

Create screencasts with menyoki

Moving Pictures

Short snippets using animated GIFs are often sufficient to show what's happening on your desktop. With menyoki, you can create these animations from the command line. BY TIM SCHÜRMANN

If you need to demonstrate complex desktop processes, a video can be worth a thousand words. Conveniently, numerous screencast tools let you record what is happening on the screen and generate a video from your actions in MP4 format. However, GIF animations are far better suited for short events. They do not display annoying blocky artifacts, can be easily sent via a messaging app, and can be quickly integrated into websites.

The menyoki [1] screencast tool lets you create animations and save the captured processes in GIF or APNG format. If so desired, menyoki can also break down videos into individual frames, which proves useful when fishing a screenshot out of a rapidly moving process or monitoring such a process in peace afterwards. Conversely, menyoki can also create an animation from individual images, making it possible, for example, to integrate photos from a wildlife camera into a short movie. In addition, menyoki lets you create screenshots and – within narrow limits – edit the individual images.

Currently, menyoki only works in an X11 session; capturing images on Wayland is not supported yet.

Installation

To get menyoki up and running, Arch Linux users can retrieve the program from the community repository by typing:

```
pacman -S menyoki
```

For other distributions, you can download the current menyoki version as a tarball from GitHub [2] and proceed to unpack the archive.

You only need the `menyoki` program in the current directory; call it by typing `./menyoki`. If you want to make menyoki available to all users, copy it to a suitable directory, such as `/usr/local/bin/`, which allows you to run menyoki directly from anywhere by typing `menyoki`. Either way, you will be responsible for updates in the future. For other installation options, see the "Alternatives" box.

And, Action!

To capture what's happening in a desktop window as a GIF animation and save it in `film.gif`, just call

Alternatives

You also can install menyoki with Docker or Cargo. Both are a bit more complicated than downloading the menyoki archive, but they make it a little easier to update the tool in the future.

If you are familiar with Docker, just call:

```
docker pull orhunp/menyoki
```

to set up the appropriate container. You can then call menyoki by typing:

```
docker run orhunp/menyoki menyoki
```

Depending on your system, you may also need to prefix the commands with `sudo` (e.g., on Ubuntu).

Menyoki also can be integrated via the Cargo package manager for the Rust programming language. All major distributions have Cargo and Rust in their repositories, but these are mostly outdated versions that are not suitable for installing menyoki. You should therefore always install the latest version of Rust manually in your home directory. To do this, first use the package manager to install the `curl` tool, a C compiler, and the developer packages for the *libX11* and *libXrandr* libraries.

On Ubuntu, line 1 of Listing 1 provides everything you need. You can download an installation script for Rust with the command in line 2. Run it, and enter `1` at the prompt. After installation, open a new terminal and install menyoki (line 3). Don't forget to regularly update Rust (line 5) and menyoki (line 6) later.

Listing 1: Setting up menyoki via Cargo

```
$ sudo apt install curl build-essential librust-x11-dev libxrandr-dev
$ curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
$ cargo install menyoki
[...]
$ rustup update
$ cargo install menyoki --force
```

Listing 2: Recording with menyoki

```
01 $ menyoki record --countdown 5 gif save "film.gif"
02 $ menyoki record --with-alpha --countdown 5 apng save "film.apng"
```

menyoki with the command shown in line 1 of Listing 2. This long line of instructions may initially look cryptic, but upon closer inspection it nicely demonstrates the menyoki command structure. Immediately following the `menyoki` command, you need to type the action that you want menyoki to complete. In Listing 2, that action, the `record` switch, records an animation.

You then use parameters to influence `record`'s behavior. In Listing 2, following `record`, the `--countdown 5` parameter tells menyoki to start recording after five seconds. This not only allows time for preparation, but – very usefully – it also lets you hide the terminal window with the `menyoki` call. Without this parameter, menyoki waits for three seconds by default. The action and its parameters are followed by the file format for storing the animation. Listing 2 uses the GIF format in line 1, but APNG is also possible (line 2).

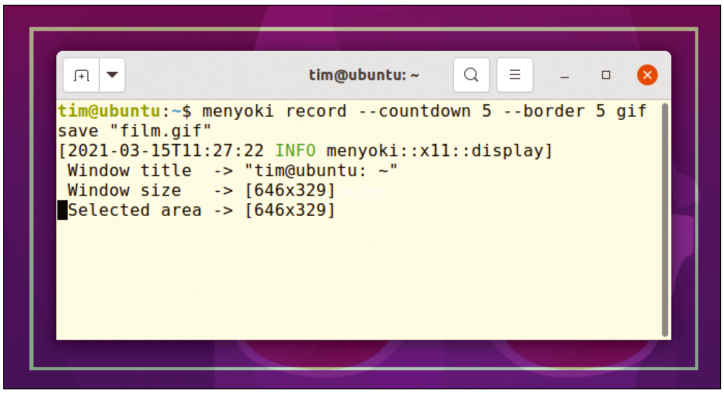
Finally, `save` tells the software to save the video in the `film.gif` file after recording. If you omit this specification, menyoki independently chooses a storage location along with the file name, which it outputs on exiting – typically `~/pictures/rec.gif`.

Framed

After calling the command from line 1 of Listing 2, a delicate frame appears (Figure 1). Menyoki will record whatever is inside the frame. You can record a different window by clicking on the desired window. In its own terminal window, menyoki informs you of the associated `Window title` and `Window size`. Armed with this information, you can determine whether menyoki has chosen the right window content. You have 60 seconds to select the desired window. If you need more time, specify the required number of seconds with the `--timeout` parameter.

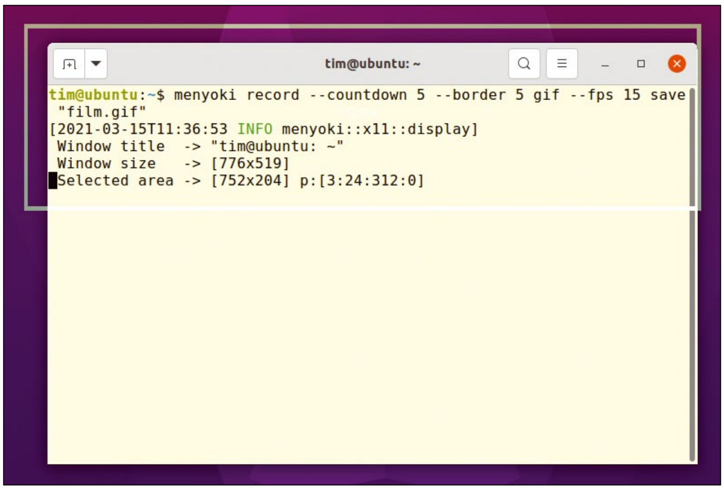
Gnome and KDE draw a shadow around their windows. Since the shadow belongs to the window, menyoki also draws it. The selection frame therefore appears to be far larger than the actual window (Figure 1). In the final animation, however, the shadow is shown as a black area.

To get around this, you have a couple of options. First, using the `--with-alpha` parameter tells menyoki also to record transparent pixels. However, this only works smoothly with the APNG format (Listing 2, line 2). Alternatively, you can hold down the Alt key and then use the arrow keys to reduce the size of the selection rectangle. You can enlarge the recording area again by holding down the left Ctrl+Alt keys. Pressing the left Shift+Alt keys along with the arrow keys lets you move the entire frame. However, exact frame placement around the window can be a fiddly process (Figure 2).



```
tim@ubuntu:~$ menyoki record --countdown 5 --border 5 gif
save "film.gif"
[2021-03-15T11:27:22 INFO menyoki::x11::display]
 Window title  -> "tim@ubuntu: ~"
 Window size   -> [646x329]
█Selected area -> [646x329]
```

Figure 1: If menyoki's selection frame seems too thin, you can expand it to five pixels, for example, using the `--border 5` parameter.



```
tim@ubuntu:~$ menyoki record --countdown 5 --border 5 gif --fps 15 save
 "film.gif"
[2021-03-15T11:36:53 INFO menyoki::x11::display]
 Window title  -> "tim@ubuntu: ~"
 Window size   -> [776x519]
█Selected area -> [752x204] p:[3:24:312:0]
```

Figure 2: By reducing the size of the selection frame, you can restrict the recording area to the areas that you want to show.

Alternatively, turn off the shadows completely in the system settings (not all desktop environments let you do this). As a final measure, you can post-process the recording with a video editing program.

Listing 3: Menyoki Options

```
01 $ menyoki record gif --fps 15 save "film.gif"
02 $ menyoki record --root --size 320x200 --padding 100:0:0:50 gif save "film.gif"
03 $ menyoki -q record --duration 1 save "-" | convert - "film.miff"
```

Recording!

After selecting the appropriate window, start recording by pressing Alt+Enter (Figure 3). The same key combination will stop the recording later. Menyoki records the video in RAM and writes it to your hard disk when you stop recording. This approach causes the main memory to quickly fill (depending on your memory size) and the system to freeze, making menyoki only suitable for short recordings.

In the default setting, the software records at 20 frames per second; however, for many tutorials, something as low as 15 could be fine. Although the results are more jerky, the finished animation also takes up less storage space. In addition, a lower frame rate gives you a longer recording time. Use the `--fps` parameter to define menyoki's recording frame rate. Unusually, this value follows the file format specification (Listing 3, line 1).

With the `--root` parameter, menyoki can capture the entire desktop. If you use multiple monitors, use `--monitor <NUM>` instead, where `NUM` corresponds to the monitor number to be recorded. Another parameter, `--focus`, lets you immediately record the currently focused window. The `--select` parameter lets you choose the window to be recorded.

If you only want to capture a section of the desktop, move the green highlight rectangle to the appropriate position using Shift+Alt and the arrow keys. Alternatively, specify the size of the area directly. For example, line 2 of Listing 3 captures a 320x200 pixel area of the desktop (Figure 4). The `--padding` switch moves the recording area 100 pixels down and 50 pixels to the right from the upper left corner of the screen.

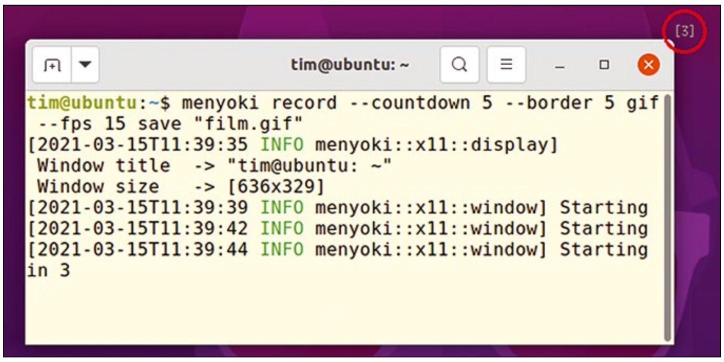
Menyoki also supports the `--duration` parameter, which lets you specify the recording time in seconds, avoiding the need to cancel recording manually. This parameter is especially useful if you use menyoki in a shell script. You can then pipe the generated animations to other programs (Listing 3, line 3).

The `-q` option tells menyoki to remain mute. The `save "-"` parameter sends the animation to standard output, which the `|` pipe then passes to `convert`. Finally, `convert` transforms the animation into the ImageMagick's MIFF format.

Cold Cuts

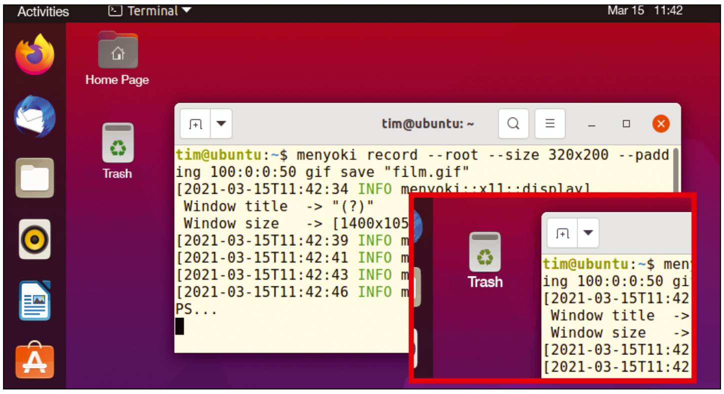
Some events happen too quickly to capture in a screenshot – for example, an important error message. To remedy this, first film the application at work and then break the animation into individual frames. Menyoki does this with the `split` action.

Line 1 of Listing 4 tells menyoki to store all frames from the video `film.gif` in JPEG format in `~/Videos/`. The file names of the frames all start with `frame_`, to which menyoki appends the frame number. Consequently, in Listing 4, you will see the files `frame_1.jpg`, `frame_2.jpg`, `frame_3.jpg`, and so on. In addition to JPG, menyoki also supports



```
tim@ubuntu:~$ menyoki record --countdown 5 --border 5 gif
 --fps 15 save "film.gif"
[2021-03-15T11:39:35 INFO menyoki::x11::display]
 Window title  -> "tim@ubuntu: ~"
 Window size   -> [636x329]
[2021-03-15T11:39:39 INFO menyoki::x11::window] Starting
[2021-03-15T11:39:42 INFO menyoki::x11::window] Starting
[2021-03-15T11:39:44 INFO menyoki::x11::window] Starting
in 3
```

Figure 3: Before recording starts, menyoki displays a countdown in the terminal, as well as in the upper right corner next to the area to be recorded.



```
tim@ubuntu:~$ menyoki record --root --size 320x200 --padd
ing 100:0:0:50 gif save "film.gif"
[2021-03-15T11:42:34 INFO menyoki::x11::display]
 Window title  -> "(?)"
 Window size   -> [1400x105
[2021-03-15T11:42:39 INFO m
[2021-03-15T11:42:41 INFO m
[2021-03-15T11:42:43 INFO m
[2021-03-15T11:42:46 INFO m
PS...
```

Figure 4: Menyoki will capture the section in the red frame with the settings shown in line 2 of Listing 3.

Listing 4: Editing Movies

```
01 $ menyoki split --dir ~/Videos/ "film.gif" jpg
02 $ menyoki edit --ratio 1.5 "frame_3.jpg" save "frame_3_big.jpg"
03 ### Scale multiple images at the same time:
04 $ for file in *.jpg; do menyoki edit --ratio 1.5 "$file" save "$file"; done
05 ### Combine individual images to create a movie:
06 $ menyoki make --dir ~/Videos/ --fps 15 --format gif save film2.gif
07 ### Pass in image files instead of directory
08 $ menyoki make --fps 15 --format gif "frame_1.jpg" "frame_2.jpg" "frame_3.jpg" save film3.gif
```

Listing 5: Making a Screenshot

```
$ menyoki capture jpg save "screenshot.jpg"
```

BMP, ICO, PNG, PNM, TGA, TIFF, and FF (farbfeld) file formats.

You can now retrieve the rapidly disappearing error message from the individual images. If the font appears too small, menyoki can enlarge the selected image directly with the `edit` action (Listing 4, line 2). The `--ratio` parameter scales the image by a factor of 1.5, preserving the aspect ratio. The results are then stored in the `frame_3_big.jpg` file.

Menyoki also supports other types of image modifications (see Table 1). Currently, however, menyoki only edits a single image and not the complete image sequence. To quickly scale all frames by a factor of 1.5, use the command from line 4 of Listing 4.

After editing all the frames, menyoki can reassemble them into an animation if so desired (Listing 4, line 6). The `--dir` parameter defines the directory with the frames, and `--fps 15` specifies a playback speed of 15 frames per second. `--format` is followed by the desired output format. Instead of your default directory, you can alternatively transfer to individual image files (line 8). The `analyze` subcommand lets you output technical information for the selected image file (Figure 5).

Table 1: Image Processing Functions

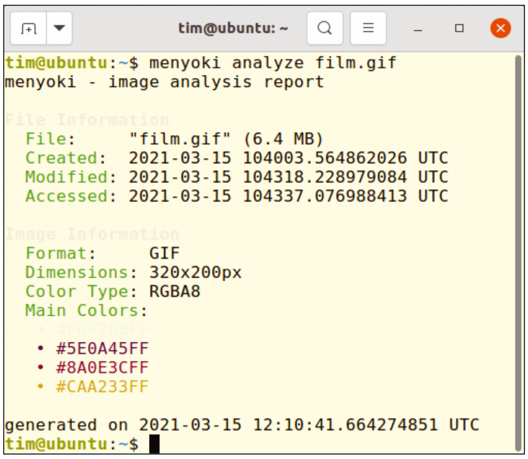
Parameters	Meaning
--grayscale	Converts image to grayscale
--crop <T:R:B:L>	Crops the image by the specified top, right, bottom, and left pixel dimensions
--resize <W>x<H>	Resizes the image by width and height without preserving the aspect ratio
--ratio <RATIO>	Resizes the image proportionally preserving the aspect ratio
--rotate <ROTATE>	Rotates the image clockwise (possible values: *90*, *180*, *270*)
--flip <FLIP>	Flips the image (possible values: *horizontal*, *vertical*)
--blur <SIGMA>	Blurs the image
--hue <HUE>	Changes the hue
--contrast <CONTRAST>	Changes the contrast
--brightness <BRIGHTNESS>	Changes the brightness

Menyoki can also create screenshots. This works in the same way as for an animation, but the associated action is `capture` (Listing 5).

Conclusions

Menyoki is the perfect tool to quickly record a process as a GIF or APNG animation. You can use menyoki to deconstruct your videos into individual frames, edit the individual images if necessary, and reassemble the image sequence into an animation. As a command-line tool, menyoki can also be easily integrated into custom shell scripts. For more examples of how to use menyoki, as well as documentation, visit the project's GitHub page [3].

However, menyoki does not yet come close to the functionality of other screencast programs. In particular, menyoki does not allow you to scale or post-process all extracted images at once. In addition, because of its mode of operation, menyoki is also only practical for short recordings. ■■■



```
tim@ubuntu:~$ menyoki analyze film.gif
menyoki - image analysis report

File Information
  File:      "film.gif" (6.4 MB)
  Created:   2021-03-15 104003.564862026 UTC
  Modified:  2021-03-15 104318.228979084 UTC
  Accessed:  2021-03-15 104337.076988413 UTC

Image Information
  Format:      GIF
  Dimensions:  320x200px
  Color Type:  RGBA8
  Main Colors:
    • #F0F2DBFF
    • #5E0A45FF
    • #8A0E3CFF
    • #CAA233FF

generated on 2021-03-15 12:10:41.664274851 UTC
tim@ubuntu:~$
```

Figure 5: Using `analyze` gives you masses of technical information about an image or animation.

Info

[1] menyoki: *https://menyoki.cli.rs*

[2] Download menyoki: *https://github.com/orhun/menyoki/releases*

[3] menyoki project page on GitHub: *https://github.com/orhun/menyoki*

Safer Internet Searches

Alternative Engines

If you are interested in data privacy, you might want to try an alternative search engine. We discuss a few search engines that serve up good results, along with an option for setting up your own search engine. BY MATS TAGE ALEXSSON

A majority of users rely on Google to find information on the Internet. Although convenient, Google can expose your personal information. Your searches result in companies tracking your online activities and then bombarding you with targeted advertising.

The privacy concerns alone might make you want to consider an alternative search engine, but there are other reasons you might consider another option. What if you want to include a search service on your personal network or website? By integrating an alternative search engine into your network or website, you can make it easier for people to find your information, while maintaining your independence from the big search giants like Google.

Luckily, there are several alternative search engines that give excellent results regardless of your motivation.

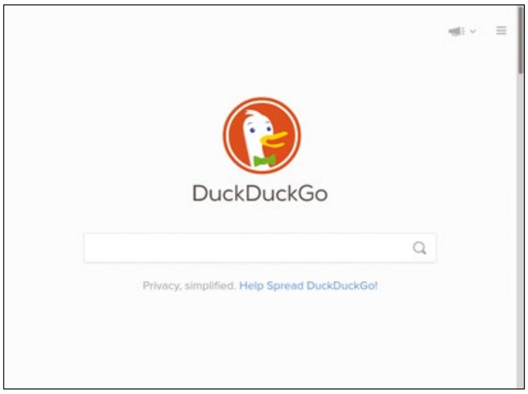
The Author

Mats Tage Axelsson is chasing around in circles trying to make his computer do more than ever.

Tracking

When it come to privacy concerns, you probably want to know how you are being tracked. Beyond knowing what data is being collected, you might also want to know how much data a company saves, how long they hold onto it, and how they use that data.

If you use Google, you can find the answers to these questions by checking your preferences. Keep in mind that many of these values are set by default, meaning you might not have selected these preferences in the first place. To find out your preferences, use Google Takeout to download your preferences and then delete all your data. Google Takeout runs in the background and sends you an email when it is done.



Privacy, simplified. Help Spread DuckDuckGo!

Figure 1: DuckDuckGo promises not to track you, store your data, or follow you around with ads.

Google also tracks you with its vast array of third-party cookies. These cookies are so unpopular that Google plans to retire them and replace them with a group profiling scheme. You can mitigate the privacy effects of these cookies by using ad blockers.

In Firefox, the privacy settings are stricter; you also can enhance Firefox with ad blockers. The Brave browser, on the other hand, blocks ads by default. Brave also lets you connect an account to your browser and get paid for advertising using Brave's own crypto currency (Basic Attention Tokens) without revealing your identity to advertisers.

Keep in mind that you can expose your IP address other ways beyond using a search engine.

To avoid tracking, your best option is to choose a search engine that doesn't track. Following are few alternatives that protect your privacy and yield good results.

DuckDuckGo

DuckDuckGo [1], the most well-known alternative search engine, does not track you (Figure 1). For instance, if you use DuckDuckGo's map function, DuckDuckGo will not even collect your position unless you activate it yourself. (Google, on the other hand, collects everything you do and can use it on all of their services.) DuckDuckGo does not even store your personal data.

In addition, DuckDuckGo offer tools to help protect your privacy. For example, a Firefox extension tells you who is tracking you and how much. The extension will tell you all the details it can find about any website you visit, as well as which websites are the worst offenders.

Qwant

Qwant [2], based in France, does their own web indexing and uses their own algorithms, creating an independent search engine (Figure 2). Working with regulators in the European Union, as well as the US, Qwant wants to help balance the competition in the search engine field.

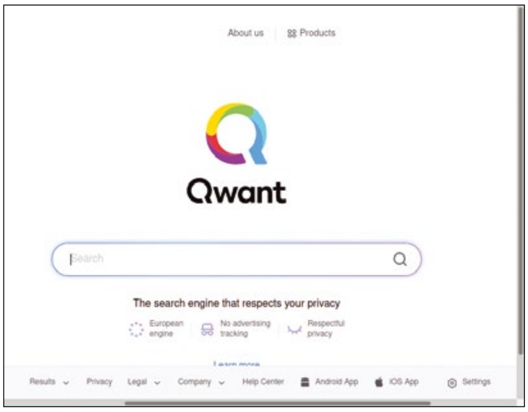


Figure 2: Based in Europe, Qwant promises to respect your privacy and not target you with ads.

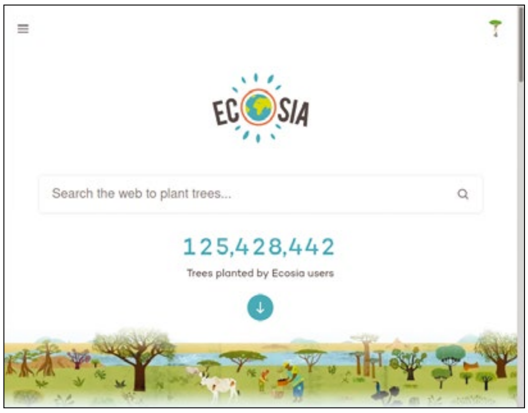

ECOSIA
Search the web to plant trees...
125,428,442
Trees planted by Ecosia users

Figure 3: Ecosia's search screen keeps a running total of trees planted.

Qwant offers products similar to those offered by Google. While Qwant does have advertisers, they don't track your every move. Features include news, images, videos, and maps (map results are sourced by Bing).

Ecosia

The Ecosia [3] search engine protects your privacy while protecting the environment. Using a portion of their profits from your searches, Ecosia plants trees (Figure 3). To see how they do this, check out their monthly financial reports online [4] .

With Ecosia, your searches are encrypted, and Ecosia does not store searches permanently. They don't sell your data to advertisers, and they don't use external tracking tools. You can even turn off tracking for the small amount of data that they do collect in order to optimize their services.

While you have full insight into everything they do, your search results are a collaboration with Bing (although you can choose between Bing and Google maps). The search results are the same as what you'd find on Bing; you just get better privacy and the satisfaction of supporting new forests.

YaCy

YaCy [5] (Figure 4), a distributed search engine, gives you the option of joining a search engine based on peer-to-peer (P2P) networking or setting up your own portal. In the P2P option, users collaboratively host the search data, but this doesn't mean you have to host your own section.

To see how YaCy works, go to YaCy's demo page [6]. While the YaCy interface is much more complex, the demo still gives you results.

YaCy's distributed nature makes the search engine more secure, plus it gives you the option to run your own instance.

Searx

Searx [7], a metasearch engine, uses other search engines' indexes to get results (Figure 5). Searx anonymizes your search request in multiple ways and the result is sent back securely to you. You can even run the search through the TOR network for more security.

Searx uses 70 search engines, most of which have an open API. For a list of your search engine choices, visit Searx's GitHub page [8].

Setting Up a Search Engine

If you want to set up a search engine for a private network or personal website, Searx and YaCy can help you get started. Both open source packages are simple to install and offer several installation methods, including Docker images, which provide an especially easy way to become familiar with each search engine as well as how the search engine affects your system.

With both Searx and YaCy, you need to pay attention to your system load. With YaCy, you also need to be mindful of your bandwidth due to YaCy's distributed nature.

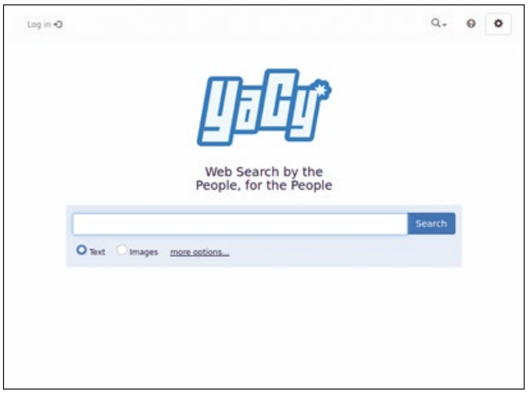


Figure 4: With YaCy, you can join a distributed web search engine or set up your own portal.

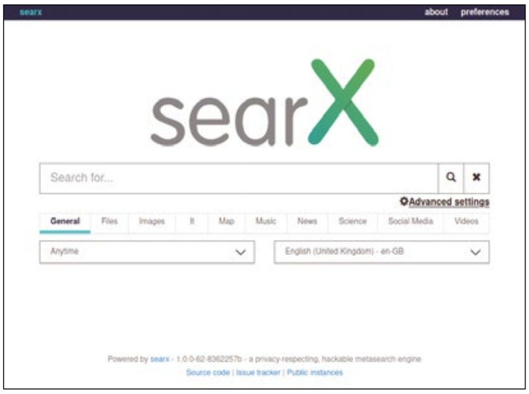

searx
about
preferences
searX
Search for...
Advanced settings
General
Files
Images
IT
Map
Music
News
Science
Social Media
Videos
Anytime
English (United Kingdom) - en-GB
Powered by searx - 1.0.0-62-8362257b - a privacy-respecting, hackable metasearch engine
Source code | Issue tracker | Public instances

Figure 5: You can customize Searx's appearance, but it starts out simply.

Both YaCy and Searx can be tested using public instances, and both offer detailed websites with information on how to change settings to match your system, as well as defaults for other settings.

Finally, Searx and YaCy are also available as source code and other binary packages. YaCy, which uses Java, is the simpler option; you run the `ant` binary to compile. However, you can also install the Searx package from your distribution's standard repository.

Searx as a Solution

If you like Docker, you can stick with it for installation. You do need to pull the Searx package and select which ports to use:

```
docker pull searx/searx
```

All Searx settings are available for both regular and Docker installations. There are advantages to using the locally compiled version with performance being one such reason. To get deeper into the installation and settings, see the Searx installation document [9].

If you want to use another installation method, download the source code from GitHub [10] and use the appropriate script to install. You'll find several scripts. The NGINX and Apache web servers are supported. If you want to add Searx to an existing site, follow the instructions on the NGINX and Apache websites.

You should also be able to find the Searx package in your distribution's repository using a general binary. This works fine unless you want to squeeze all the performance possible out of your system.

Listing 1: Setting secret_key

```
server: port : 8888
bind_address : "127.0.0.1" # address to listen on
secret_key : "SuperSecretKey" # change this!
base_url : <http://localhost:/> # Set custom base_url.
Possible values: False or
"<https://your.custom.host/location/>"
image_proxy : False # Proxying image results
through searx http_protocol_version : "1.0" # 1.0 and
1.1 are supported
method: "POST" # POST queries are more secure as
they don't show up in history but may cause problems when using
Firefox containers
default_http_headers:
X-Content-Type-Options :nosniff
X-XSS-Protection : 1;
mode=block
X-Download-Options : noopen
X-Robots-Tag : noindex, nofollow
Referrer-Policy : no-referrer
```

Searx Settings

Regardless of installation method, you will use the default settings file (`settings.yml`) to set your preferences. You can use the default options, which contains many well-known search engines and some lesser known ones.

You also need to set up a directory for `settings.yml` and then point to it. The files in this directory are the ones controlling Searx. Since the settings file is written in YAML, you have access to all of YAML's features for creating links (like adding a site) that work as searches for your service.

When you compile, you will need most expected libraries for handling networking, the build, etc. You should check out the `uwsgi` install, which is a minimal binary protocol for communicating between the nodes.

In the main settings, you have one absolutely vital task to perform: setting the `secret_key` value (Listing 1). To choose this value, your best option is to let OpenSLL create it for you, but you can also use a password manager. The other options under the main settings are useful but not necessary to get started.

If you look through the rest of the settings file, you will find the valid search providers for your instance. You can also use this file to limit your searches if you have concerns about a particular provider.

To add more providers (you can even add a single page), you need to add a section to the settings file. Copy the default file and edit what you need. As a simple example, you can edit the Wikipedia settings entry in the file:

```
- name : wikipedia engine : ↩
  wikipedia shortcut : wp base_url :
  'https://{language}.wikipedia.org/'
```

In addition to the `language` variable, you'll find the `query`, `page`, and `params` variables, which can be used to control your searches. You also can control the type of results that are returned. Result options include strings, images, and videos, as well as torrent files.

YaCy as a Solution

One of the biggest differences between YaCy and Searx is that YaCy runs independently of other search engines. YaCy creates its own distributed index. Just like in torrent files that use distributed hash tables (DHTs), you keep your own part of the tables.

To run YaCy, you need to set the amount of space that you will allow YaCy to occupy on your system, although the installation script has a default. Like Searx, you can use a Docker image to run YaCy. YaCy offers three different Docker images: `amd64`, `arm64v8`, and `arm32v7`.

To install YaCy with Docker, use the standard values found on YaCy's web page:

```
docker run -d --name yacy -p ↩
  8090:8090 -p 8443:8443 -v ↩
  yacy_data:/opt/yacy_search_server↩
  /DATA --log-opt max-size=200m↩
  --log-opt max-file=2 ↩
  yacy/yacy_search_server:latest
```

These standard values help you manage resource usage. Once the server is running, you can also access a management interface from your browser. If you want to be able to use the management interface from another computer, you need to set an administrator password. If you lose the password, you will need to go back to the command line in the root of the YaCy directory and run:

```
bin/password.sh
```

This command will handle changing the password, whether your server is running or not.

You can also clone the GitHub repository and compile the binaries [11]. Confusingly, the GitHub repo does not mention at the top that you must compile before running the standard script (`startYACY.sh`).

YaCy needs Java. When you download the GitHub repo, you need `ant` to compile. You'll find the details further down in the GitHub document. If you need to install YaCy on multiple machines, you can create a Debian package directly with the compiler.

Configuring YaCy

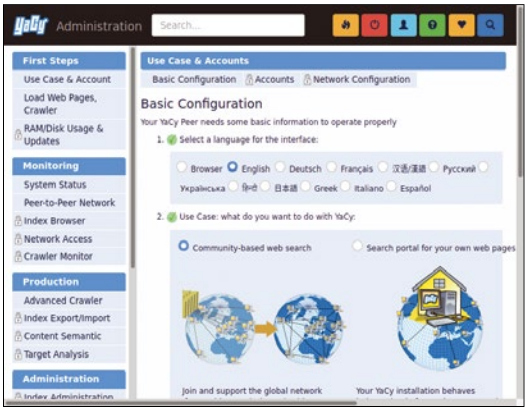
Whichever method you choose for installation, you need to set up some values to get the most out of your system. First, you should specify how you want to use YaCy. For the most basic configuration, you set an interface language, name, and search use case (Figure 6).

The search use case sets the type of search. An internal search will just find files on your network; more common is a search of the entire YaCy community.

In the YaCy Administration dialog, you can edit all your settings, including working memory, disk space, and more.

Clicking on *RAM/Disk Usage & Updates* lets you adjust the settings for working memory and disk space. The default memory for the Java Virtual Machine (JVM) is set to 600MB.

The other values in the RAM/Disk Usage & Updates dialog save you from running out of disk space. You can use the Steady-state minimum option to disable crawls when free disk space falls below a specified minimum megabytes. This will only be an issue when you have the ports open and you collaborate with the index or when you start your own crawl. HTCache configuration lets you control the size of the content retrieved via HTTP or FTP; the default size is 4GB.



Basic Configuration

Your YaCy Peer needs some basic information to operate properly

1. Select a language for the interface:

○ Browser ● English ○ Deutsch ○ Français ○ 汉语/漢語 ○ Русский ○ Українська ○ हिन्दी ○ 日本語 ○ Greek ○ Italiano ○ Español

2. Use Case: what do you want to do with YaCy:

● Community-based web search ○ Search portal for your own web pages

Join and support the global network

Your YaCy installation behaves

Figure 6: Under Basic Configuration, you can setup your interface language and use case.

Putting YaCy to Work

Once you've configured YaCy, you can start a crawl from any web address. From the Administration dialog, click on *Load Web pages, Crawler* and enter the web address. YaCy will look through all the documents on the server and index them for you. You can use this to index your own internal network or add your new web page to the common index.

In addition to private searching, YaCy lets you share your search engine with others. You can customize YaCy for your website. Click on *Portal Configuration* to set color, title text, and even the logo that appears above the search box. From here, you also can see what the search engine will look like with your customizations.

If you use YaCy seriously, you should consider contributing to the YaCy index. To do this, you need to open your port to other peers on the network. In particular, you'll need to open port 8090, which is usually blocked by default.

Conclusion

If you care about privacy, you should consider using one of the alternate search engines discussed in this article. Along with a better search experience, you can protect your personal data and maybe even the environment.

With YaCy and Searx, you can even set up a personal search engine for your home or office. Searx lets you spread your search over many services, while making your search anonymous. Searx takes some work to set up, but you can achieve a standard install in less than an hour. If you want to go all out, you can join the YaCy network and index as much as your disk space and CPU can handle. The search results are not necessarily as good as commercial solutions, but you will probably find what you need.

Regardless of which search engine you choose, you can rest easy knowing your personal information is protected. ■■■

Info

[1] DuckDuckGo: *https://duckduckgo.com/*

[2] Qwant: *https://www.qwant.com/*

[3] Ecosia: *https://www.ecosia.org/*

[4] Ecosia financial reports: *https://blog.ecosia.org/ecosia-financial-reports-tree-planting-receipts/*

[5] YaCy: *https://yacy.net*

[6] YaCy demo page: *https://yacy.net/demonstration_tutorial_screenshot/*

[7] Searx: *https://searx.space/*

[8] Searx search engines: *https://searx.github.io/searx/admin/engines.html*

[9] Searx installation document: *https://searx.github.io/searx/admin/installation.html*

[10] Download Searx: *https://github.com/searx/searx.git*

[11] YaCy on GitHub: *https://github.com/yacy/yacy_search_server*

WIREGUARD Secure and simple VPN tool
WEBCAMS AND LINUX
LINUX MAGAZINE
ISSUE 237 – AUGUST 2020
WEBCAMS AND LINUX
Smoothing out hardware wrinkles with Guvcview and QtCAM
Text-to-Speech in LibreOffice
Guacamole Remote access
GARDEN TRICKS USE LONG-RANGE RADIO TO MONITOR YOUR PLANTS
BUILD AN IRC BOT
ISSUE 239 – OCTOBER 2020
BUILD AN IRC BOT
Customize a GTK3 Theme
DISPATCHER SCRIPTS Automate NetworkManager for seamless location changes
Kernel Lockdown Mode Rein in root with this powerful new Linux security feature
Build a WiFi Sound System With a Rasp Pi and speakers from IKEA
RHEL 8.2 What's new in Red Hat's flagship distro
Write a Script To find old
CALL IN WITHOUT LOCK-IN Get connected with the Jitsi videoconferencing tool
SPEED UP YOUR SYSTEM HOT TWEAKS FOR A FASTER LINUX
ISSUE 238 – SEPTEMBER 2020
SPEED UP YOUR SYSTEM
Hot tweaks for a faster Linux
PSI Discover the kernel's cool new performance monitoring feature
Steampunk Laptop A little copper tubing and a Raspberry Pi
LINUXVOICE
hub: A command-line tool for managing your GitHub repository
maddog: Remembering the Linux/Alpha project
PulseEffects: Equalizer and effects for PulseAudio
Livestream Your Backyard Birdhouse
SMARTER DIRECTORIES
ISSUE 236 – JULY 2020
SMARTER DIRECTORIES
Reinventing the file storage metaphor with semantic tagging
OpenCart Build a secure online store
Design an Ebook with Free Tools
Monitoring Tricks Build a dashboard to keep watch on your old Linux computer
Water Your Plants Tending your garden with a Rasp Pi Zero
ProjectLibre Organize and visualize your next project
bauh All-in-one tool for managing Flatpak and AppImage packages
20 YEARS ARCHIVE DVD
HUGE SAVINGS! £29.90 VALUE!
20TH ANNIVERSARY ISSUE! ALL 239 ISSUES ON A SEARCHABLE DISC!
20 YEARS
ALIVE!
usql Manage PostgreSQL, MySQL, and SQLite with a single interface
Mistborn All-in-one tool for protecting your LAN
JAM.PY BUILD A WEB FRONT END FOR YOUR DATABASE
SECURE YOUR SYSTEM
ISSUE 241 – DECEMBER 2020
SECURE YOUR SYSTEM
Expert tips for protecting your Linux
Clonezilla SE Mass cloning without the mess
Managing Servers with Cockpit
Christmas Tinkering Build a holiday music box
Git Utilities Cool tools for extending Git version control
Fun with Go Display song lyrics line by line
IOBROKER Home automation with a Rasp Pi
CHOOSE A SHELL
ISSUE 245 – APRIL 2021
CHOOSE A SHELL
We compare Bash, Zsh, and fish
ELK Stack Keep watch over your logs
Rendering Images as Text
Boop Keep it local with this scriptable notepad
Xubuntu on a Dell Refurbish your old laptop
detLFS Learn about Linux by building a mini OS
Go Tricks Create a Maze
LINUXVOICE
The rise of the small Internet
Font Manager: Keep your type straight
maddog: The benefits of a POS/ERP system
FOSSPicks LibreSprite Pixel Editor, NeoChat Chat Client, Drumstick MIDI Monitor
Tutorial Efficient Searches with ugrep
WWW.LINUX-MAGAZINE.COM
GO PROGRAMMING Search for files on Google Drive
3D PRINTING
ISSUE 242 – JANUARY 2021
3D PRINTING
MOFO Linux Steer around censorship with this privacy-focused distro
Nyttig Share files and stream Internet radio from a Rasp Pi
Netdata Zero configuration monitor
Nerf Dart Game Score points for your next neighborhood
Stacer: Clean up your system
maddog: Morality and licensing
JessyInk: Presentations in
FIREFOX Enabling hardware acceleration
Exploring the iNet Wireless Daemon
ISSUE 243 – FEBRUARY 2021
iNet
Intel's wireless daemon ushers in a new world for Linux networks
Sunflower New start for a classic twin-panel file manager
Optimizing Vector Graphics Speed up your websites with these tricks for scrunching SVG files
Extensions in Tiny Core
Escape Room Build a cool electronic game using an Arduino
MS Visual Studio on Linux You can even develop code for a Rasp Pi
SAFE MESSAGING PASS YOUR PUBLIC KEY USING DNS
ISSUE 244 – MARCH 2021
STREAM PROCESSING
New coding techniques for the data revolution
Make It Mandatory Requiring two-factor authentication at login
2020: The Year in Review
Kconfig Deep Dive Exploring the powerful system that configures the Linux kernel
Raspberry Pi 4 on 8GB RAM Do more and go faster with the souped-up 8GB version
Checksecurity Lock down your system with automated security checks
FREE DVD
Double-Sided DVD INSIDE!

MakerSpace
Video Calls on a Rasp Pi
LINUX MAGAZINE
Faster Startup
Tune your system for speedier boot
Top Tip: Squeeze more time out of a battery charge with auto-cpufreq
FOSS TOOLS REVIEWED!
FREE DVD
MakerSpace
Hands-On Guide to the GPIO
LINUX MAGAZINE
Post-Quantum Encryption
Will privacy survive quantum computers?
JSON Deep Dive
Get fluent in this important data format
DTLS
Encryption over UDP
LINUXVOICE
• maddog explores the history of text editors
• Obsidian: Nifty note and knowledge tool
10 FOSSPICKS
HEAVENLY FREE TOOLS!
WWW.LINUX-MAGAZINE.COM

FOSSPicks

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

Open source has finally made it to Graham's kitchen this month, where he's installed four speakers connected to a Raspberry Pi channel amplifier and the wonderful HiFiBerry Linux distribution. BY GRAHAM MORRISON

UX designer

Penpot

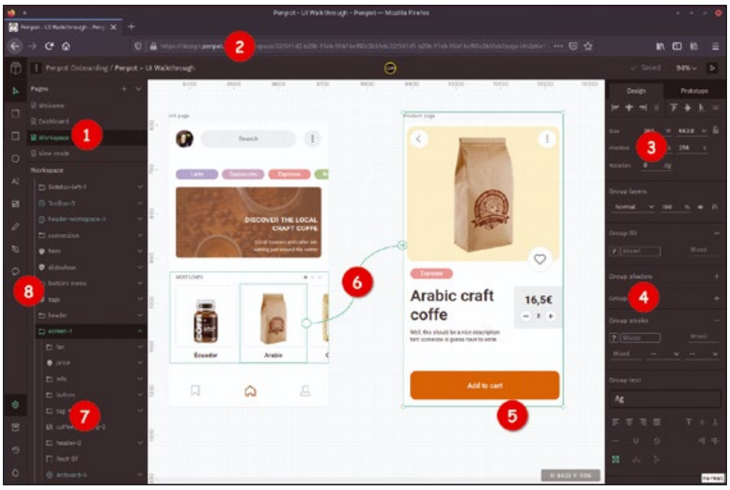
Penpot is a tool that helps designers create and prototype a user interface (UI). A little different from the desktop and command-line utilities that typically fill these pages, Penpot is a web application, designed to run in your web browser and, most usefully, online. Like WordPress, Nextcloud, and Home Assistant, the online element in Penpot is an essential feature that performs such an important role that the application is worth investigating, regardless of how it presents itself. And like those other brilliant web apps, it has a huge Linux bias and a commitment to open source.

Over the past few years, if you worked on any project that required a designer, it's likely the designer will have used a prototyping tool like Penpot. They're used to sketch up various UI designs, present them to the team, make changes, and eventually fix those changes so that developers can start adding whatever functionality is required. This is why similar prototyping tools work best when they're hosted online: It allows a project's various "stakeholders" to access the designs, make their own suggestions, and leave feedback. Penpot can do all of this and a lot more.

While Penpot is a web app and its developers offer their own portal, which is currently free to use, the open source version can be easily downloaded and self-deployed via a Docker image. This also happens to be the default development environment for anyone who wants to tinker with the code. Despite loading via a web browser, the application itself operates much like Inkscape or Scribus, only with additional social and prototyping features. There are shape drawing tools, a freehand mode, paths, curves, images, and text boxes. All are tightly bound to a grid, object boundaries, many alignment modes, and accessible data values for every element. These can then be grouped into reusable objects.

The UI is remarkably quick and responsive. As you might expect from a tool for designers, it really has been beautifully designed. From the icons and expandable menus to the comprehensive example documents that are used to onboard you to the workflow and functionality, you soon forget you're using a web application. In common with other user experience (UX) tools, when you're done with design, it's also possible to build some skeleton functionality into your creations to see how it feels to interact with your proposed interface. This is the prototyping element, and it lets you link one element to another so that transitions can be made with a click. It's simple but likely to be developed further in the future, considering that Penpot is still alpha software.

When it comes to sharing your ideas, you can create teams to allow other users to access specific projects, and projects are chosen from a dashboard before being loaded into a workspace, which is where the creating and editing happens. The open source element in Penpot is obviously important too, not just because one of the first panels you're presented with invites you to get involved with the code development, but also because Penpot makes fantastic use of the SVG format. Penpot uses it for import and export, along with wide browser compatibility and performance. Using SVG means you're never going to be locked into a single application.


Penpot - UI Walkthrough - Penpot — Mozilla Firefox
Penpot Onboarding / Penpot - UI Walkthrough
Pages
Workspace
Design
Prototype
Search
DISCOVER THE LOCAL CRAFT COFFE
Ecuador
Arabic
Arabic craft coffe
16,5€
Add to cart
1
2
3
4
5
6
7
8

1. Dashboard and workspaces: Projects are opened from the dashboard into the workspace. 2. Hosted: Host your own project and access Penpot from a browser. 3. Properties: Exact values for every element can be edited. 4. Assets: Easily access styles, assets, buttons, and properties. 5. Canvas: Zoom in and out, group elements, and easily arrange them. 6. Prototyping: Create one-click actions to see how the application might feel and behave. 7. History: Every action is saved and can be recalled or reset. 8. Drawing tools: Forget you're in a browser and draw just like you're using Krita or Inkscape.

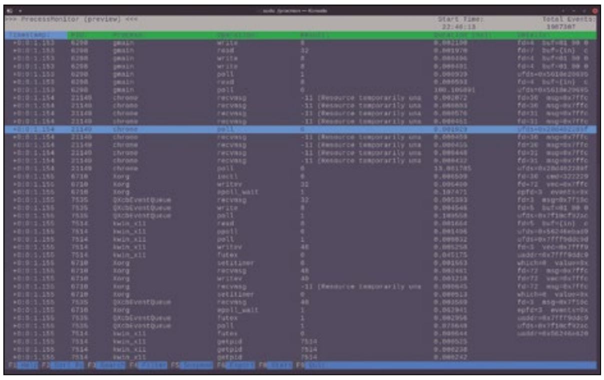
Project Website
https://penpot.app/

System monitor

ProcMon

If you've ever had the pleasure of using Microsoft Windows, you might have run a Microsoft tool called Process Manager (ProcMon) to find out why your system is running slowly. Unlike other monitoring tools that might show you which processes were consuming the most resources, ProcMon focuses on which system calls (syscalls) a process is currently working on, enabling you to perform some analysis on what is going on behind a simple process name. Syscalls represent the functions that an application or process has requested to be run by the operating system or kernel, such as reading or writing data, allocating memory, waiting (polling) for an interrupt or a message, or opening a device. Monitoring which syscalls a process is running is a great way to troubleshoot your own projects or to determine whether a random process might be failing, stalling, or performing badly. It's more granular than the process-wide analysis offered by tools like `top` or a desktop system monitor.

Microsoft has ported its famous ProcMon to Linux and placed the source code on GitHub from where it can be easily built and executed. There's even a command-line option to build a deb package. ProcMon Linux runs within a terminal using curses and presents much of the same information as its desktop Windows counterpart. Most importantly, this means showing the resource-hungry syscalls, the process IDs they're sharing, the operations they're performing, and how much of your system time they're consuming. It's a fascinating insight into what various applications might be doing, such as Chrome, X11, or your Windows compositor, and it shows a different side to resource usage than the normal `top` kind of view. With specific command-line arguments, you can choose specific process IDs to trace, track only specific syscalls, and output a log of traces to an external file, which can be later reread and run through for analysis. ProcMon is definitely a refreshing change from normal Linux resource monitors and a useful addition to your `$PATH`.



On Windows, ProcMon might be ideal for identifying malware. On Linux, it's perfect for feeling smugly free of malware.

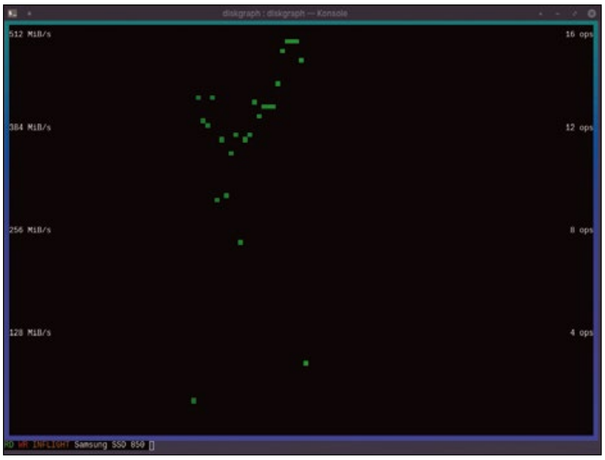
Project Website
https://github.com/Sysinternals/ProcMon-for-Linux

Storage monitor

diskgraph

These days, there are too few purely C-coded utilities that are useful, modern, self-contained, compiled in seconds, and relatively easy to understand. However, diskgraph is an instantly compiled command-line tool that draws a chart in your terminal showing storage throughput for both input and output operations. Each measurement is represented by a character-sized square that represents a 500ms interval, with the *y*-axis showing an automatically scaled bandwidth (amount transferred over time) and the *x*-axis showing the time frame. The blocks will scroll from the right to the left as time progresses so you can more easily see fluctuating load conditions as you read and write files, allowing you to easily see periods of peak and low transfer and how your hardware is performing. It's very simple, but it presents information that isn't easily visualized any other way.

The executable takes only a single argument, the storage device, with no further options to configure how the utility looks or behaves. But this is an ideal opportunity rather than a limitation, because the entire project's code is contained within a single `.c` file of just less than 500 lines of code. This makes it ideal for study and also for user-developed modification such as extra arguments. It would be trivial to change the character I/O blocks into bars, for instance, or to add some other visualization method or to change the way the bandwidth is calculated. It's exactly the kind of project to start with if you want to investigate more serious system-level Linux programming, maybe ultimately leading even to kernel development. The code is relatively easy to understand, operates at a low level, and should be safe to modify – diskgraph only reads from devices rather than writing to them without requiring special privileges.



Diskgraph is a super simple tool for monitoring storage I/O, but it's also eminently hackable and easy to understand.

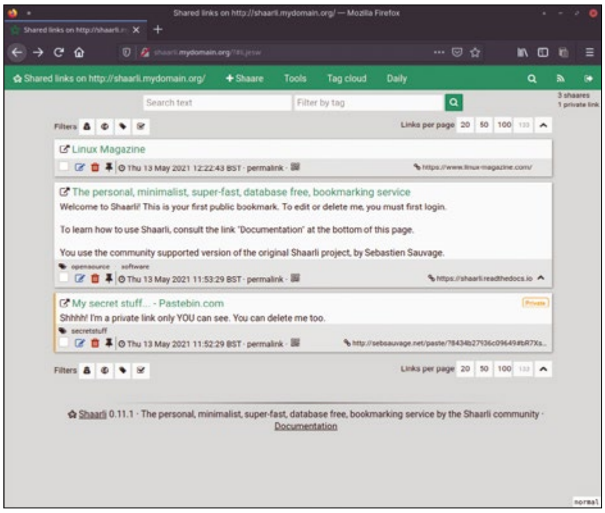
Project Website
https://github.com/stolk/diskgraph

Bookmark manager

Shaarli

There was a time, many years ago, when managing web browser bookmarks was incredibly important. This was before Google got good, and you could seldom find the site you'd previously found or remember how you got there. Instead, you'd fruitlessly hunt through your browser history trying to remember the train of thought that led you to the site you wanted to revisit. If you had the foresight to bookmark the site, the problem was solved until you bookmarked too many sites and failed to organize them, leading you back to the rapidly improving Google. But bookmarks haven't gone away, especially now with many of us experimenting with other search providers and trying to keep as much personal preference off the Internet as possible. Consequently, it can still be a good idea to control and manage your own bookmarking regime.

We recently looked at a brilliant tool and local bookmark server called `buku`, but Shaarli has a slightly different take on the same problem. Rather than relying on a system command to handle the bookmarks and a server to present them, Shaarli handles everything from a web server using PHP. Installation is straightforward if you already have a server running, and the documentation excellently guides you through creating a virtual host in your configuration. There are also instructions for using Docker. When the service is up and running, you simply create an account, log in, and start adding bookmarks. The great thing about Shaarli is that these bookmarks can (optionally) be public and appear listed on the landing page so that anyone with access to your server can see them. Thanks to the search and tagging, a daily feed and even RSS provision, Shaarli can become a hub of activity and a central part of using the Internet outside of the big search and bookmark providers. Locked behind a firewall, Shaarli could be a useful local sharing hub for your family and friends.


Shared links on http://shaarli.mydomain.org/ — Mozilla Firefox
Shared links on http://shaarli.m
shaarli.mydomain.org/?#LjesW
Shared links on http://shaarli.mydomain.org/
+ Shaare
Tools
Tag cloud
Daily
Search text
Filter by tag
3 shaares
1 private link
Filters
Links per page
20
50
100
133
Private
normal

Linux Magazine

Thu 13 May 2021 12:22:43 BST · permalink · https://www.linux-magazine.com/

The personal, minimalist, super-fast, database free, bookmarking service

Welcome to Shaarli! This is your first public bookmark. To edit or delete me, you must first login.

To learn how to use Shaarli, consult the link "Documentation" at the bottom of this page.

You use the community supported version of the original Shaarli project, by Sebastien Sauvage.

openasource software

Thu 13 May 2021 11:53:29 BST · permalink · https://shaarli.readthedocs.io

My secret stuff... - Pastebin.com

Shhhh! I'm a private link only YOU can see. You can delete me too.

secretstuff

Thu 13 May 2021 11:52:29 BST · permalink · http://sebsauvage.net/paste/?8434b27936c09649#bR7Xs...

Shaarli 0.11.1 · The personal, minimalist, super-fast, database free, bookmarking service by the Shaarli community · Documentation

While we wouldn't recommend running Shaarli on the Internet, it's a great way to save and share bookmarks on a home network.

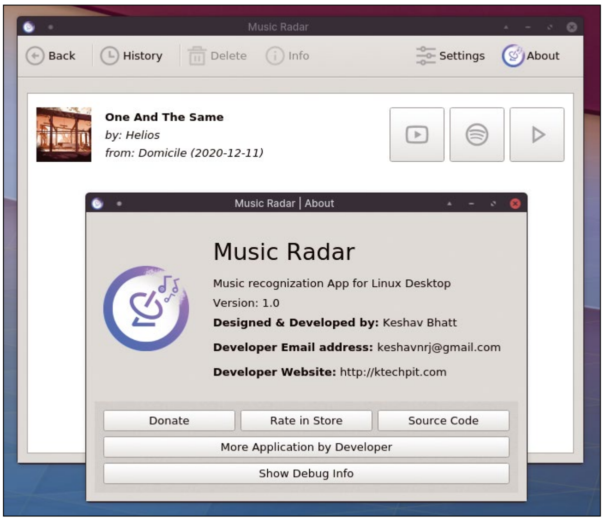
Project Website
https://github.com/shaarli/Shaarli

Music identifier

Music Radar

Even though we're used to the idea of a tool that can listen to music and identify it, especially on a smartphone, the reality is often different from the idea. Background noise, poor recording quality, and limited access to the clip you want to identify all inhibit the ability of these tools to do their job. As a result, there was a genuine "jaw-dropping" moment when we first tested out Music Radar on a piece of music. The piece was a rather obscure ambient track from 2020 by the artist Helios entitled "One And The Same." It has no drums, no melody, no repeating sections, no verse, and no chorus. It consists mainly of textured soft clusters of droning synthesizer chords and, at times, sounds almost identical to Brian Eno's 1983 classic, "Deep Blue Day." Music Radar listened to about seven seconds of "One And The Same" and perfectly identified it, proudly presenting the result with a thumbnail of the album cover, the title, artist, and year, along with links to search YouTube, play it on Spotify, and play a preview – seriously impressive.

This power has a price, though. The analysis and results come via API access to *audd.io*, which permits only a few requests per day. Music Radar isn't affiliated with this service, but you can create a token for yourself by creating an account and pasting the key into the Settings pane. This window also allows you to set the default recording time and which theme to use. Thanks to some clever choices, the application's main functionality couldn't be easier to use. There are two options for sourcing the audio: a connected microphone or whatever you're using to listen to music, such as a speaker. A speaker allows Music Radar to capture the audio at the highest quality, unaffected by external noise or microphone input, and surprisingly requires no messing around with PulseAudio. However it does this, regardless of the online AI magic it must perform, it just works.



One And The Same
by: Helios
from: Domicile (2020-12-11)

Music Radar
Music recognization App for Linux Desktop
Version: 1.0
Designed & Developed by: Keshav Bhatt
Developer Email address: keshavnrj@gmail.com
Developer Website: http://ktechpit.com

Music Radar is brilliant if you listen to Internet radio and need to identify a track.

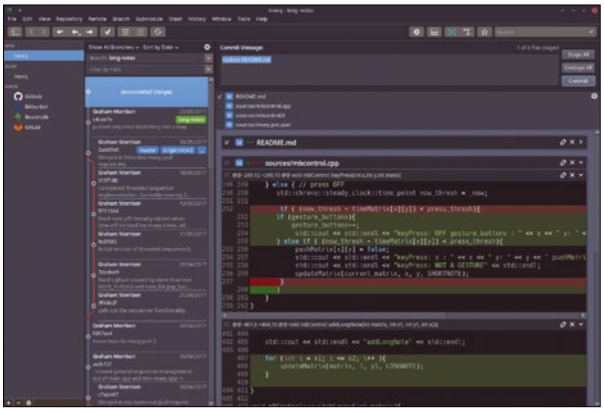
Project Website
https://github.com/keshavbhatt/music-radar

Git GUI

GitAhead

Most of us agree that Git is wonderful, and it's now being used everywhere, even in places not routinely handling code. Documentation, design, and even configuration files can often benefit from using Git, which can create a problem when non-technical people need to wrestle with some of Git's more esoteric features. This is where a graphical interface can help by promoting best practices, safe-guarding from mistakes, and by simply visualizing normally abstract commands. The problem is, we've yet to find a Git GUI that's able to do all of this without becoming more complex than the `git` command itself. But GitAhead is a very strong contender, especially in one area of Git's functionality: commit history. Understanding and visualizing a Git repository's commit history can help you to understand exactly what's happening when you later create and merge pull requests, or release branches, or even resolve conflicts with the `git` command itself.

GitAhead excels at this. It will connect directly with remote repositories on GitHub, Bitbucket, Beanstalk, and GitLab or access local repositories directly. These are saved into a quick access panel on the left and, by default, the last commit is shown in the main view when a repository is selected. The main view shows either an inline `diff` of the changes between this version and the previous, or a tree view of the file structure holding those changes. In all views, a middle pane displays a navigable log of the commit history. Each commit is clearly shown along a colored timeline for each branch, making it easy to see what came from where. You can also create a new branch, squash, merge, stash, checkout, and rebase from the menu system. It works brilliantly, takes much of the guesswork out of managing a Git repository, and is also a beautiful example of interface design.



```
} else { // press OFF
    std::chrono::steady_clock::time_point now_thresh = _now;

     if ( (now_thresh - timeMatrix[x][y]) < press_thresh){
    if (gesture_buttons){
        gesture_buttons++;
        std::cout << std::endl << "keyPress: OFF gesture_buttons : " << x << " y: " <
    } else if ( (now_thresh - timeMatrix[x][y]) < press_thresh){
        pushMatrix[x][y] = false;
        std::cout << std::endl << "keyPress: x : " << x << " y: " << y << " pushMatri
        std::cout << std::endl << "keyPress: NOT A GESTURE" << std::endl;
        updateMatrix(current_matrix, x, y, SHORTNOTE);
    }
    }
}
}
```

```
std::cout << std::endl << "addLongNote" << std::endl;

for (int i = x1; i <= x2; i++ ){
    updateMatrix(matrix, i, y1, LONGNOTE);
}

}
```

Visualize branches and commits easily, as well as pull, merge, and rebase, with the easy-to-use GitAhead Git client.

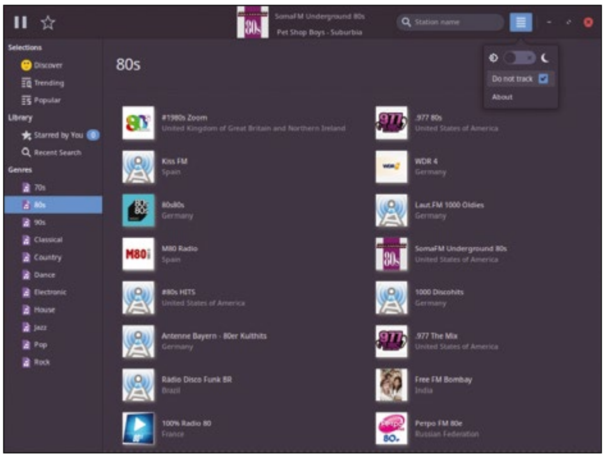
Project Website
https://gitahead.github.io/gitahead.com/

Internet radio client

Tuner

Maybe it's because many of us have been dreaming about traveling and experiencing other cultures, but there's been a resurgence in high-quality desktop Internet radio applications. We recently looked at the fantastic Shortwave, for example, which made tuning into a global network of radio stations easy, and here we have the equally wonderful Tuner. Tuner isn't designed specifically for global radio, but it can access a huge library of available stations through its search function, along with quick access to a variety of genres, trending stations, and a discovery tab. It describes itself as a "minimalistic" Internet radio application, so it's fitting that Tuner makes such brilliant use of the GTK3+ toolkit and its human interface guidelines to reduce the screen clutter to only those buttons you need.

The top window border contains nearly all the playback functionality. It embeds the play, pause, and star (for favoriting a station) buttons. To the right of these are the station details, a search field, and a tiny configuration menu. The main window below is populated with either search results or curated selections from the genre list. The station URLs themselves are retrieved online using the API and catalog on *radio-browser.info*, but unlike with Shortwave, this can be changed by setting an environment variable. This allows you to host your own list of stations, for instance, or use more than one source for searches and recommendations. The application was developed as a COVID-19 project by its developer after moving from macOS to Linux, and this helps explain why its design is so clean, minimal, and functional. In fact, the project has proven so successful that the developer has been overwhelmed by the response and requirements of the community and is now looking for developers and package maintainers to help. If you're interested, take a look at the project's GitHub page.



Tuner has one of the best Gnome interfaces we've seen; it is a brilliant way to explore the global Internet radio network from your desktop.

Project Website
https://github.com/louis77/tuner

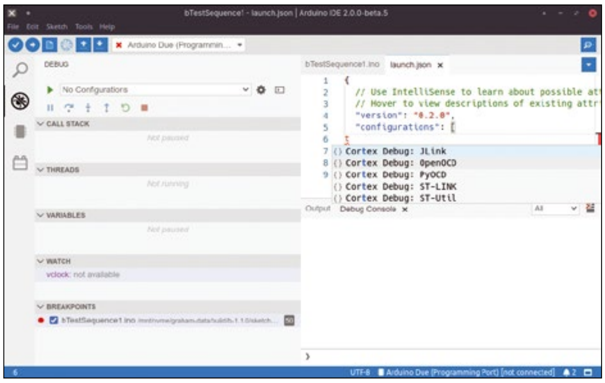
Development environment

Arduino IDE 2

The original Arduino IDE started life in 2005, before the launch of Twitter and before Facebook dropped its university-only invite policy. This text editor based IDE has since become synonymous with Arduino devices, being used for everything from uploading simple sketches (really, C files) to device management and fully fledged application development. At its heart, however, it was always a text editor with a lower panel for debug and compilation output, with few features to help with larger projects. This is why some projects created their own makefiles to automate the build, and partly why the `arduino-cli` tool was developed to unshackle the build process from the editor. More recently, `arduino-cli` has led to some excellent integration with Visual Studio (VS) Code, which can now develop, debug, build, and flash Arduino projects to the hardware better than the original Arduino IDE.

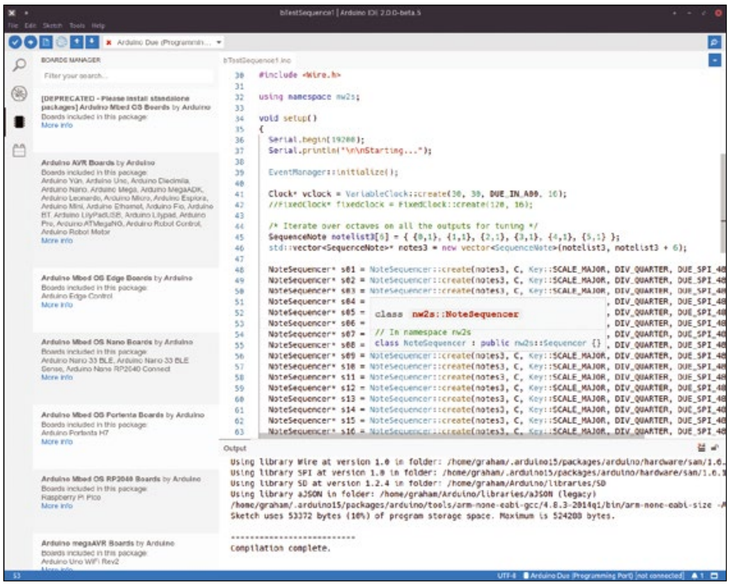
This means now is the time for an ambitious update to the humble Arduino editor, and the Arduino team has already been working on it for a couple of years. Gone is the Java dependency, gone is the one-window-per-file view, and gone is debugging via printing string messages. It's all been replaced by Arduino IDE 2, a modern Eclipse Theia and Electron-based front end to all the decoupled functionality of `arduino-cli`, transforming the default Arduino experience. This unification on a single build platform, `arduino-cli`, means you can choose whichever tool works best for you and easily migrate between them as your needs change. Edit from the IDE one day and build from VS Code the next.

The biggest difference is in the editor itself. It now uses an adaptive design that will scale to fit whatever space you want to give it. It feels similar to VS Code without the complexity or Microsoft telemetry, which is no surprise because they both use the same architecture. There are autocomplete suggestions as you type, allowing you to easily navigate between symbols and definitions in your code and refactor names at will. A left-hand pane includes tabs that show search results, the board manager, the library manager, and a debugging console. But it's this debugging console that represents the other biggest change in functionality, because it finally drags Arduino development into the 21st century. Debugging can now be performed "live," in real time, allowing you to run your code on a board while viewing its execution within the IDE.



```
{
  // Use IntelliSense to learn about possible at
  // Hover to view descriptions of existing attr
  "version": "0.2.0",
  "configurations": [
}
{} Cortex Debug: JLink
{} Cortex Debug: OpenOCD
{} Cortex Debug: PyOCD
{} Cortex Debug: ST-LINK
{} Cortex Debug: ST-Util
```

Debugging is no longer reliant on printing serial output but can instead use two-way communication while your code is executing on the device.


bTestSequence1 | Arduino IDE 2.0.0-beta.5
File
Edit
Sketch
Tools
Help
Arduino Due (Programmin...
BOARDS MANAGER
Filter your search...
bTestSequence1.ino
Output
53
UTF-8
Arduino Due (Programming Port) [not connected]
1

[DEPRECATED - Please install standalone packages] Arduino Mbed OS Boards by Arduino
Boards included in this package:
More info

Arduino AVR Boards by Arduino
Boards included in this package:
Arduino Yún, Arduino Uno, Arduino Diecimila, Arduino Nano, Arduino Mega, Arduino MegaADK, Arduino Leonardo, Arduino Micro, Arduino Esplora, Arduino Mini, Arduino Ethernet, Arduino Fio, Arduino BT, Arduino LilyPadUSB, Arduino Lilypad, Arduino Pro, Arduino ATMegaNG, Arduino Robot Control, Arduino Robot Motor
More info

Arduino Mbed OS Edge Boards by Arduino
Boards included in this package:
Arduino Edge Control
More info

Arduino Mbed OS Nano Boards by Arduino
Boards included in this package:
Arduino Nano 33 BLE, Arduino Nano 33 BLE Sense, Arduino Nano RP2040 Connect
More info

Arduino Mbed OS Portenta Boards by Arduino
Boards included in this package:
Arduino Portenta H7
More info

Arduino Mbed OS RP2040 Boards by Arduino
Boards included in this package:
Raspberry Pi Pico
More info

Arduino megaAVR Boards by Arduino
Boards included in this package:
Arduino Uno WiFi Rev2
More info

```
#include <Wire.h>

using namespace nw2s;

void setup()
{
  Serial.begin(19200);
  Serial.println("\n\nStarting...");

  EventManager::initialize();

  Clock* vclock = VariableClock::create(30, 30, DUE_IN_A00, 16);
  //FixedClock* fixedclock = FixedClock::create(120, 16);

  /* Iterate over octaves on all the outputs for tuning */
  SequenceNote notelist3[6] = { {0,1}, {1,1}, {2,1}, {3,1}, {4,1}, {5,1} };
  std::vector<SequenceNote>* notes3 = new vector<SequenceNote>(notelist3, notelist3 + 6);

  NoteSequencer* s01 = NoteSequencer::create(notes3, C, Key::SCALE_MAJOR, DIV_QUARTER, DUE_SPI_48
  NoteSequencer* s02 = NoteSequencer::create(notes3, C, Key::SCALE_MAJOR, DIV_QUARTER, DUE_SPI_48
  NoteSequencer* s03 = NoteSequencer::create(notes3, C, Key::SCALE_MAJOR, DIV_QUARTER, DUE_SPI_48
  NoteSequencer* s04 =                                                   , DIV_QUARTER, DUE_SPI_48
  NoteSequencer* s05 =   class  nw2s::NoteSequencer                      , DIV_QUARTER, DUE_SPI_48
  NoteSequencer* s06 =                                                   , DIV_QUARTER, DUE_SPI_48
  NoteSequencer* s07 =   // In namespace nw2s                            , DIV_QUARTER, DUE_SPI_48
  NoteSequencer* s08 =   class NoteSequencer : public nw2s::Sequencer {} , DIV_QUARTER, DUE_SPI_48
  NoteSequencer* s09 = NoteSequencer::create(notes3, C, Key::SCALE_MAJOR, DIV_QUARTER, DUE_SPI_48
  NoteSequencer* s10 = NoteSequencer::create(notes3, C, Key::SCALE_MAJOR, DIV_QUARTER, DUE_SPI_48
  NoteSequencer* s11 = NoteSequencer::create(notes3, C, Key::SCALE_MAJOR, DIV_QUARTER, DUE_SPI_48
  NoteSequencer* s12 = NoteSequencer::create(notes3, C, Key::SCALE_MAJOR, DIV_QUARTER, DUE_SPI_48
  NoteSequencer* s13 = NoteSequencer::create(notes3, C, Key::SCALE_MAJOR, DIV_QUARTER, DUE_SPI_48
  NoteSequencer* s14 = NoteSequencer::create(notes3, C, Key::SCALE_MAJOR, DIV_QUARTER, DUE_SPI_48
  NoteSequencer* s15 = NoteSequencer::create(notes3, C, Key::SCALE_MAJOR, DIV_QUARTER, DUE_SPI_48
  NoteSequencer* s16 = NoteSequencer::create(notes3, C, Key::SCALE_MAJOR, DIV_QUARTER, DUE_SPI_48
```

```
Using library Wire at version 1.0 in folder: /home/graham/.arduino15/packages/arduino/hardware/sam/1.6.
Using library SPI at version 1.0 in folder: /home/graham/.arduino15/packages/arduino/hardware/sam/1.6.1
Using library SD at version 1.2.4 in folder: /home/graham/Arduino/libraries/SD
Using library aJSON in folder: /home/graham/Arduino/libraries/aJSON (legacy)
/home/graham/.arduino15/packages/arduino/tools/arm-none-eabi-gcc/4.8.3-2014q1/bin/arm-none-eabi-size -A
Sketch uses 53372 bytes (10%) of program storage space. Maximum is 524288 bytes.

--------------------------
Compilation complete.
```

One brilliant thing about the new Arduino IDE is that you can finally change the font size and theme without messing around with configuration files and CSS.

For this to work, you do need one of the more recent boards (MKR boards, Nano 33 IoT, Nano 33 BLE, Portenta, and Arduino Zero devices) or a bridging device such as the SAM-ICE JTAG to interface. This is because debugging now uses two-way communication while executing your code, a process that was previously impossible. In the old IDE, after the code had been uploaded to the board, the only interaction between your device and the IDE was through the serial connection, often leaving developers to debug their code using `println` statements. The new process allows you to introspect the state of your application, including its variables and functions as they're lighting up LEDs, scanning inputs, and producing voltages, just as you do when debugging software running on your computer. All of this hopefully sets up the next generation of the Arduino IDE for the next 15 years of development, in that useful junction between simple editor and fully fledged IDE, and we can't wait to see what features they add next.

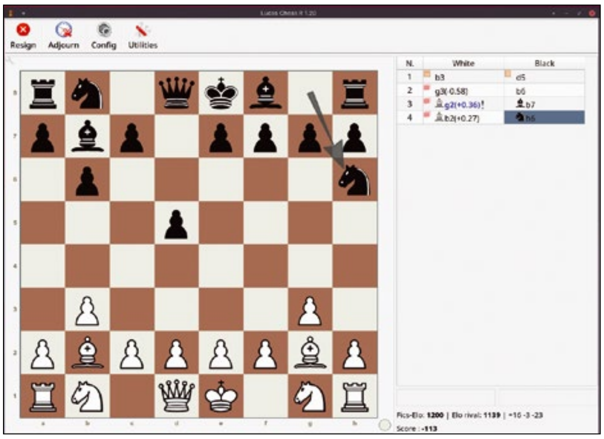
Project Website
https://github.com/arduino/arduino-ide

Chess simulator

Lucas Chess

It's been more than 20 years since IBM's Deep Blue defeated a reigning (human) world chess champion, and chess has since been through various troughs and peaks in popularity. But thanks to recent circumstances, and a particularly popular episodic streaming production, the humble game of chess is going through a huge renaissance. It's become difficult to even buy a decent chess board. Fortunately, there are plenty of computer chess engines and clients to choose from on Linux, including the widely regarded and ported PyChess with the Stockfish back end. But one of the most popular open source implementations has never had official support on Linux, until now. And that's the brilliant Lucas Chess. What makes Lucas Chess so popular, despite its antiquated graphics and long development periods, is the quality of the chess it plays. This isn't because it plays a particularly strong game of chess, although it can – Deep Blue and its descendents have also solved that problem – but because it plays a natural game of chess that can feel like playing a human. Alongside this, Lucas Chess also includes some of the best training modes to help you improve your game, which are difficult features to find on Linux.

Lucas Chess includes 61 different playing engines, all with differing levels of difficulty, from 0 to 3300 in the Elo difficulty rating scale. There are even special engines for younger players, with names like Monkey, Lion, and Stephen, and there are the brilliant training modes using the Stockfish or Komodo engines. You can pit your skills against specific challenges or receive scored feedback and analysis of every move you make. There are memory games, coordinate tests, a mode to help learn by repetition, and training positions, plus other tests and starting and ending resources for stronger players. The user interface is simple but includes a huge list of themes that can change the board, colors, and piece designs. It may all look a little austere, but then chess is old.



N.	White	Black
1	b3	d5
2	g3(-0.58)	b6
3	♗g2(+0.36)!	♝b7
4	♗b2(+0.27)	♞h6

Fics-Elo: 1200 \| Elo rival: 1139 \| +16 -3 -23
Score : -113

Until now, if you wanted to play Lucas Chess on Linux, you needed to use Wine, but there's finally a version that runs natively.

Project Website

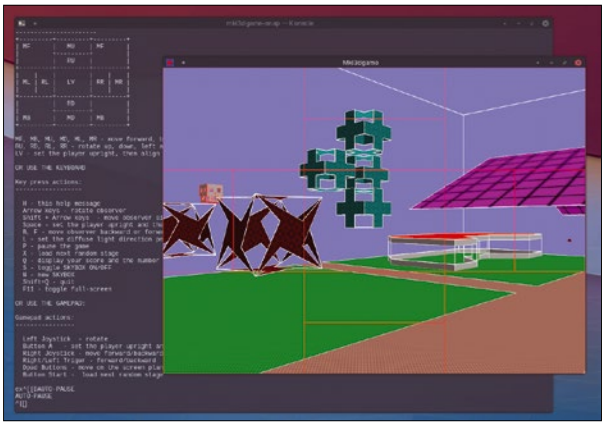
https://lucaschess.pythonanywhere.com/home

Finding game

mki3dgame

Smartphones, iPads, and Android tablets have rekindled popularity for all kinds of old-fashioned games, and in particular, games that involve drawing, coloring, and finding things. The weirdly named mki3dgame is a finding game, but rather than constraining itself to the limitations of a two-dimensional plane, mki3dgame hides objects for you to find within a 3D realm. Its name actually comes from the MKI 3D Rapid Modeler, a web-based, open source, and keyboard-driven 3D modeling tool, which was used to create all the 3D shapes in the game. The idea behind the game itself is to find and frame multiple red tokens hidden within the 3D environment using the targeting reticule to center the token and the direction keys to move toward the token so that you collide to collect it.

The command-line output of the game helpfully lists the various keyboard controls for moving the player through the 3D space. It also reports when a token is collected and how many more you have to find. There are also keyboard shortcuts for changing the graphics and loading a new random level. Sections of the targeting reticule can be clicked on to move or rotated without needing a keyboard, gamepad, or joystick, which makes the game a fun distraction. It can be quite difficult, though, as you move through the levels trying out different angles and approaches to reveal the tokens, and it's a process that's made more difficult by multiple monsters also moving through the environment, ending your attempt if you make contact with them. The latest release of the game includes four different types of these monsters and 32 stages to complete, which makes finishing the whole thing a considerable challenge. But if you enjoy finding things and want to improve your spatial awareness from a 2D dimensional screen, it's definitely worth the installation.



```
MF, MB, MU, MD, ML, MR - move forward, 
RU, RD, RL, RR - rotate up, down, left 
LV - set the player upright, then align

OR USE THE KEYBOARD

Key press actions:
------------------

  H - this help message
  Arrow keys - rotate observer
  Shift + Arrow keys  - move observer 
  Space - set the player upright and th
  B, F - move observer backward or forw
  L - set the diffuse light direction 
  P - pause the game
  X - load next random stage
  Q - display your score and the number
  S - toggle SKYBOX ON/OFF
  N - new SKYBOX
  Shift+Q - quit
  F11 - toggle full-screen

OR USE THE GAMEPAD:

Gamepad actions:
----------------

  Left Joystick  - rotate
  Button A    - set the player upright a
  Right Joystick - move forward/backwar
  Right/Left Triger - forward/backward
  Dpad Buttons - move on the screen pla
  Button Start -  load next random stag

ex^[[IAUTO-PAUSE
AUTO-PAUSE
^[[
```

Apart from the game itself, mki3dgame is a great project to study 3D and Golang programming.

Project Website

https://github.com/mki1967/mki3dgame

Keep your documents organized with SeedDMS

Paper Pusher

The SeedDMS document management tool helps your team stay focused and offers some powerful features for search, notification, and process control.

BY MARCO FIORETTI

A Document Management System (DMS) is a file manager on steroids: a tool that stores files in orderly fashion but with many more features than a normal file manager provides. A team can use a DMS to co-manage documents of all sorts, from invoices to courseware, books, and product specifications, coordinating all their work on the document through an organized workflow.

SeedDMS [1] is a multiplatform, GPL-licensed DMS tool that provides version control, metadata search, and sophisticated workflow management. This tutorial will help you get started with managing documents in SeedDMS.

Why a DMS?

Before I plunge into the details of installing and using SeedDMS, I'll take a moment to address the question of why a DMS is even necessary. The first thing that any DMS gives its users is finer control – of both privacy and security. Total separation between DMS accounts and system user accounts also makes it much easier and safer to add temporary "guests" – customers or interns who need access to internal documents without full access to the system.

Support for metadata like keywords, categories, comments, and arbitrary attributes, together with the possibility to search both the metadata and the actual content of each document, allows users to organize and retrieve their files in many different ways.

Above all, a DMS administrator can decide who should review or approve each class of documents, allowing for a coherent, constant process that each document must pass through before it is officially approved. A DMS makes teamwork easier and team members accountable in ways that are not possible with normal users accounts and shared folders on a file server.

Why SeedDMS

One advantage of SeedDMS is that you can try it without installing anything. The SeedDMS website includes an online demo [2]. Your only limit is that you cannot upload files bigger than 5MB. If you have Docker credentials, you could also start your own test instance of SeedDMS inside an online virtual environment such as Play with Docker [3].

Feature-wise, SeedDMS has the basic functions found on most DMS tools, such as access control lists for both users and groups, plus the possibility to adopt a custom or built-in workflow for reviewing, approving, and publishing documents. The user and administration interfaces are entirely web-based, available in multiple languages, and compatible with WebDAV servers. Thanks to its integration with the Lucene search engine, SeedDMS offers full text search for files in PDF, Word, and Excel formats.

Another interesting feature of SeedDMS is its readiness for multi-site operations: You can install one copy of the source code and then share it among many different, totally independent instances. This makes it easy for consultants, or organizations whose departments have very different ways of working, to give each of their customers a SeedDMS configuration that is optimal for their needs. The same feature also allows each of the organizations that might share the same server to migrate to new versions of SeedDMS with minimum effort and impact on users.

Architecture

SeedDMS itself is a PHP7 application. A working installation is a mix of six components: web server, database, basic PHP libraries, PHP PEAR Modules, assorted utilities like ImageMagick or pdftotext for image and text processing, plus the actual PHP code of the application.

If you run any common, well-supported Linux distribution, you either have most of the background components already, or, if SeedDMS asks for them, you can install them from the default repositories. The PHP core of SeedDMS is divided into two independent parts. One is the back end, packaged as a PHP PEAR module, that accesses the database. The other is the front end that generates the web interface and handles user input.

Installing SeedDMS

SeedDMS installation could be documented and integrated better than it is. It surely helps to have some previous experience with installing LAMP (Linux + MySQL + Apache + PHP) applications in Linux, but either way, the process is still manageable without particular pain. You'll need to go through the following phases, which are quicker than they look:

1. Prepare a database (optional).
2. Unpack the code archive in the correct folder.
3. Create a flag file and set access permissions.
4. Edit the configuration file.
5. Configure PHP and the web server.
6. Finish the configuration from the web interface.

SeedDMS can use the serverless SQLite, as well as the MySQL/MariaDB or PostgreSQL database engines. At this time of writing, support for PostgreSQL is officially considered less finished than the other options. SQLite is the easiest to adopt because it is directly supported by PHP. However, at least within SeedDMS, it is slower and less robust than the others if your installation must handle more than very few users or a few hundred files.

To make SeedDMS run with MySQL or MariaDB, you must first create a dedicated database and user for it, then populate that database with the right tables. The steps for creating a database are well documented [4], so I will not describe them here. You can add the tables to the newly created database by typing the following command:

```
#>  cat SEEDDMS/seeddms/↩
install/create_tables-innodb.sql ↩
| mysql -u SEEDDMS_USER -p SEEDDMS_DB
```

where `create_tables-innodb.sql` is the file that contains all the necessary MySQL commands, and the three uppercase strings are placeholders for the base directory of your SeedDMS installation, the name of the database it must use, and its dedicated user.

The SeedDMS distribution includes similar instruction files for the SQLite and PostgreSQL databases, and you can also perform the same operations in the graphical interface of Step 6 by ticking one box.

Installing SeedDMS

The SeedDMS website hosts several releases of SeedDMS versions 5 and 6. Unless you need to customize SeedDMS or run it with a heavily customized PHP configuration, it is highly recommmended to download and use only the compressed tar files that have a "quickstart" suffix. Those archives contain all the SeedDMS code, plus all the PHP PEAR modules it needs to work, as well as a complete directory tree that will also serve as a template for shared installations. The screenshots and instructions in this tutorial come from the 6.0.15 "quickstart" version of SeedDMS, installed on an Ubuntu 20.04 Linux system with the Apache web server, using the following commands:

```
#> wget seeddms-quickstart-↩
6.0.15.tar.gz
#> tar xf seeddms-quickstart-↩
6.0.15.tar.gz
#> sudo cp -r -p seeddms60x ↩
/var/www/html/dms
```

After putting all the SeedDMS files in a folder where they will be reachable by your web server, move to that folder and run the following commands (assuming your web server user is `www-data`):

```
#> chown -R www-data www-data data
#> chown www-data www-data conf/↩
settings.xml
#> touch conf/ENABLE_INSTALL_TOOL
```

The first two instructions make the configuration file and the data folder of SeedDMS writable by the web server. The third creates an empty "flag" file that authorizes the graphical install tool to work.

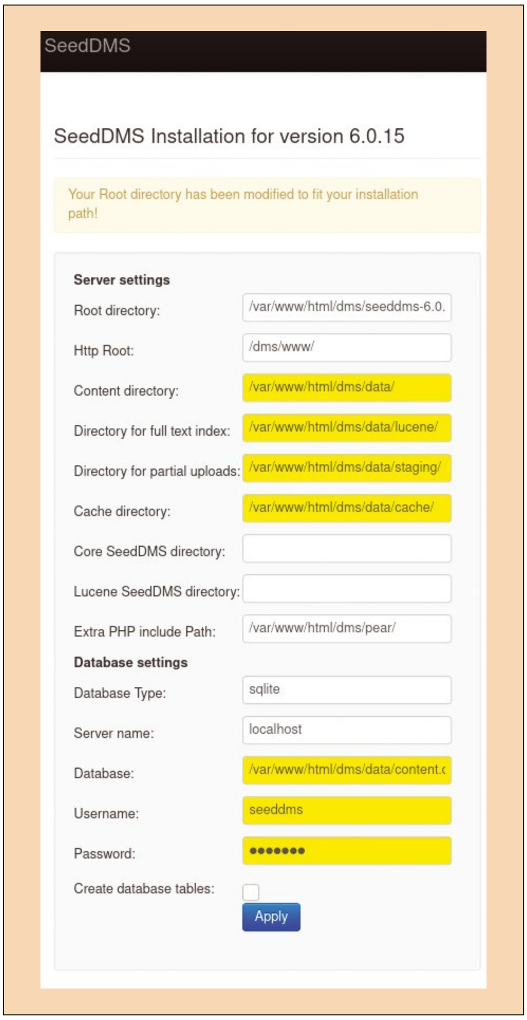
The SeedDMS configuration file `settings.xml` contains plenty of configurable variables, together with documentation of what they do. Unless you have special needs, however, you will only need to change one or two settings before moving to the next step. One is `server rootDir`, which must point to the SeedDMS subfolder inside your SeedDMS root. Because I had copied all the files into `/var/www/html/dms`, I set this variable to:

```
server rootDir=↩
"/var/www/html/dms/seeddms/"
```

The other variable that might need editing is `database db_driver`, which specifies the database to use. However, as you can see in Figure 1, you could pass this information to SeedDMS from the graphical administration interface the first time you log in as administrator.

PHP and Web Server Configuration

Even the "quickstart" versions of SeedDMS depend on several general-purpose PHP packages. If they are not already present on your Linux server, the graphical interface will ask you to install them. On Ubuntu 20.04, for example, I



SeedDMS

SeedDMS Installation for version 6.0.15

Your Root directory has been modified to fit your installation path!

Server settings

Root directory: /var/www/html/dms/seeddms-6.0.

Http Root: /dms/www/

Content directory: /var/www/html/dms/data/

Directory for full text index: /var/www/html/dms/data/lucene/

Directory for partial uploads: /var/www/html/dms/data/staging/

Cache directory: /var/www/html/dms/data/cache/

Core SeedDMS directory:

Lucene SeedDMS directory:

Extra PHP include Path: /var/www/html/dms/pear/

Database settings

Database Type: sqlite

Server name: localhost

Database: /var/www/html/dms/data/content.c

Username: seeddms

Password: •••••••

Create database tables: □

Apply

Figure 1: The final phase of a SeedDMS installation, where you can specify database and search engine parameters.

had to install the following packages from the command line: m

```
#> sudo apt install php-xml ⮐
php7.4-gd php-mbstring php-pdo-sqlite
```

For security reasons, only the `www` subfolder of a SeedDMS installation must be directly accessible from web browsers. Therefore, if you are running an Apache web server as I did, you should first set the `DocumentRoot` variable for your SeedDMS in your Apache configuration file to the complete path to that folder (`/var/www/html/dms/www` in my case). Next, you should also enable the `rewrite` and `headers` modules of Apache with the following commands, and then restart the server:

```
#> sudo a2enmod rewrite
#> sudo a2enmod headers
#> sudo systemctl restart apache2
```

These commands will make Apache load and use the security settings, written in several `.htaccess` files shipped with SeedDMS, that block unwanted direct access to your data.

Log In and Finish

At this point, you can point your browser to the base URL of your SeedDMS installation, log in as `admin` with password `admin`, and finish the configuration. The first thing to do, after changing the admin password, is to pass SeedDMS all the information about the database you want to use, as shown in Figure 1.

Before you start to use SeedDMS, you should also configure an automatic backup procedure. The backup should include the configuration file `conf/settings.xml`, the whole data folder, and unless you chose SQLite, also the whole MySQL or PostgreSQL database. On a related note, please note that you should tick the *Create database tables* box of Figure 1 only when you are actually creating a whole new instance of SeedDMS; otherwise, the tables you had backed up will be rewritten!

Users and Permissions

Once your SeedDMS is up and running, and you are logged in as administrator, select *Admin Tools* from the top menu to load the control panel of Figure 2. One of the easiest and most useful things to do right away could be to define global keywords, categories, and custom attributes for the documents that your team will actually manage. In addition to the global list managed by the administrator, each user can have her own list of keywords. Figure 3 just hints at how much customization is possible: Basically, you have all you need to catalog your documents exactly as you wish. The main risk is to indulge these features so much that the system becomes too overwhelming to use!

The parameters in Figure 4 give an idea of how it is possible to monitor the activity and workload

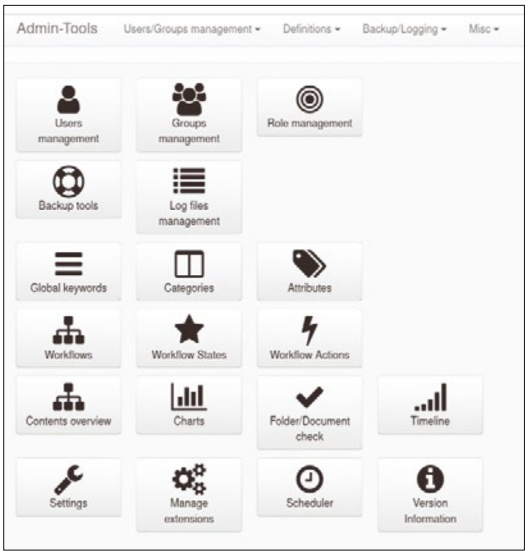
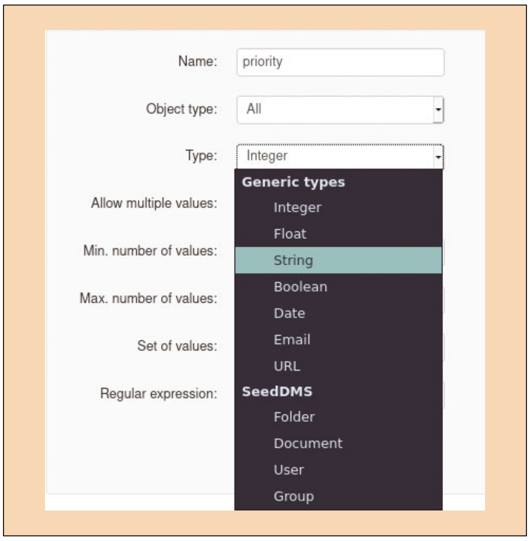


Figure 2: The main control panel of a SeedDMS administrator.



Name: priority

Object type: All

Type: Integer

Allow multiple values:

Min. number of values:

Max. number of values:

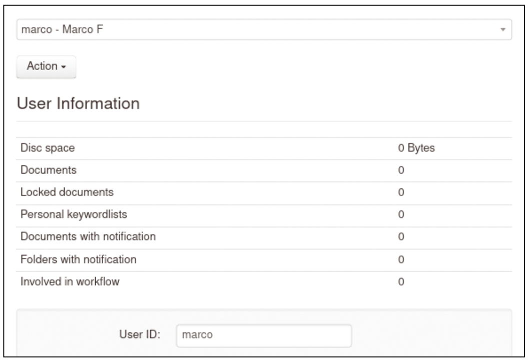
Set of values:

Regular expression:

Figure 3: If owner, timestamps, and other ordinary metadata aren't enough, you can create your own inside SeedDMS.

of every SeedDMS user. Another tool available to managers to plan and monitor team activities is visible in Figure 5: a timeline that shows present and future deadlines of all documents plus many other parameters, from their revision status to the number of attachments.

Each folder or document can have separate default permissions, as many revisions as needed, and much more. Personally, what I like in SeedDMS is links: Every user with editing rights over some documents can define links, that is, direct relationships among the documents.



marco - Marco F

User Information

Disc space	0 Bytes
Documents	0
Locked documents	0
Personal keywordlists	0
Documents with notification	0
Folders with notification	0
Involved in workflow	0

User ID: marco

Figure 4: Activities, workload, and resource usage of every SeedDMS user are immediately visible to the administrator.

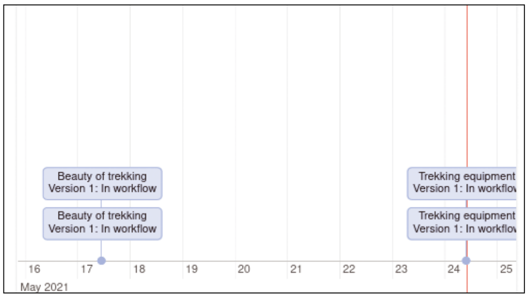


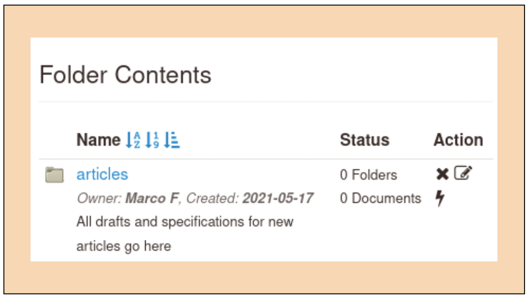
Figure 5: The SeedDMS timeline window can show all the documents being processed and their advancement status.

Users and Workflows

The predefined roles of admin, user, and guest should be enough in almost all cases because any administrator can designate substitutes and give each user access to only the minimum necessary folders and workflows.

The user interface for adding and managing documents is quite spartan, but it is also relatively self-explanatory and easy to use. Figures 6 and 7 show how a document looks inside SeedDMS and some of the things you can do with it. Once you upload a file, you can set its expiration date inside SeedDMS, as well as add keywords, categories, comments, and version information. You can also configure which users or groups should be notified when work is done on the document.

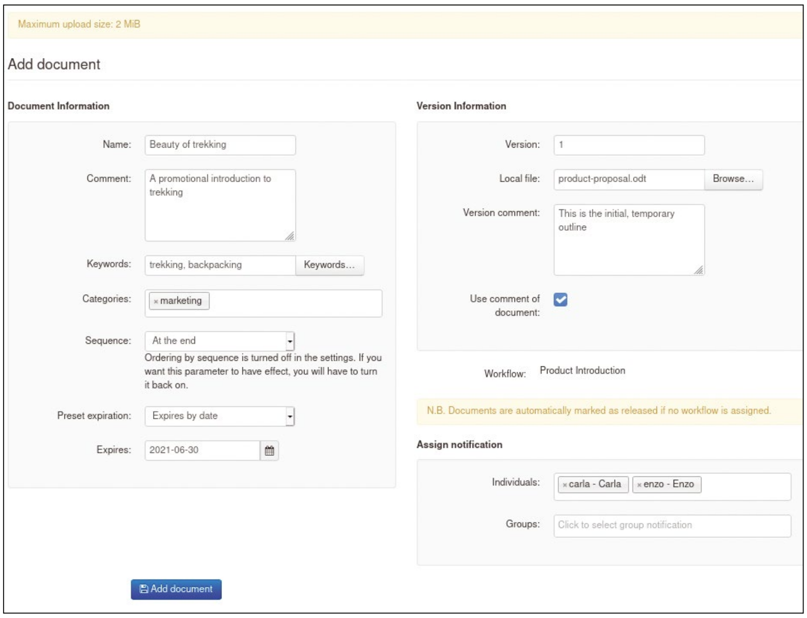
The good part of SeedDMS is that once you have configured the right workflow, using it is easy, productive, and well worth the effort. Indeed, workflows are the main reason to use a DMS. The bad part is that designing a workflow inside SeedDMS takes time and requires some discipline, brainstorming, and testing. No single, pre-packaged workflow will be good for everybody, because every organization has unique needs and constraints.



Folder Contents

Name	Status	Action
articles *Owner:* ***Marco F****, Created:* ***2021-05-17*** All drafts and specifications for new articles go here	0 Folders 0 Documents	✖ ✎ ϟ

Figure 6: All files and folders inside SeedDMS have a clear label that specifies their purpose.



Maximum upload size: 2 MiB

Add document

Document Information

Name: Beauty of trekking

Comment: A promotional introduction to trekking

Keywords: trekking, backpacking

Categories: × marketing

Sequence: At the end

Ordering by sequence is turned off in the settings. If you want this parameter to have effect, you will have to turn it back on.

Preset expiration: Expires by date

Expires: 2021-06-30

Version Information

Version: 1

Local file: product-proposal.odt

Version comment: This is the initial, temporary outline

Use comment of document: ☑

Workflow: Product Introduction

N.B. Documents are automatically marked as released if no workflow is assigned.

Assign notification

Individuals: × carla - Carla × enzo - Enzo

Groups: Click to select group notification

Figure 7: The main Document Information window of SeedDMS, where each user can set and update all the corresponding information.

Another reality of every document management system is that nobody, including you, will use it if it is too cumbersome or time-consuming. In other words, you must avoid the risk that a SeedDMS workflow forces your team to adapt to it, rather than the workflow adapting to the team.

The secret is to build a strong understanding of the workflow before you start using SeedDMS. Start with a pen, paper, and real observations of your team at work. Evaluate different options, and consider that different groups inside your organization might need different workflows. Once you are really confident that you have defined the workflow you and your peers really want (or need …), implement it in SeedDMS.

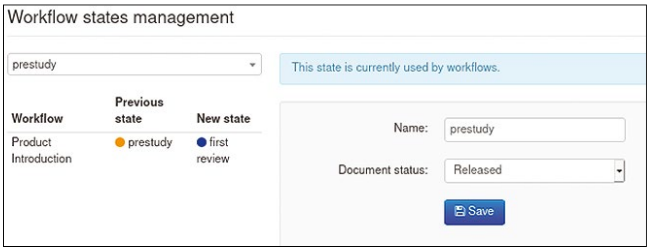
SeedDMS makes it possible to design custom workflows. For me, the most efficient way to go is to start by creating all the necessary states and actions. In Figure 8, for example, I created a *prestudy* state, in which the status of a document is "Released."

The next step is to connect all the states and actions inside SeedDMS, just as they are on paper. Clicking on the *Workflows* button opens the interface of Figure 9: The left half contains a graphical, dynamic representation of your workflow that SeedDMS updates as soon as you add or remove states. You can change the layout of that diagram (not the connections!) as you want by dragging its elements. Remember to click on *Save Layout* when you are done.

The right side of Figure 9 is where the real translation of your workflow design from paper to software happens. On the top, you specify the name and first state of your workflow. The form at the bottom is the place to add the transitions from any state to any other, the action that make them happen, and which users can perform those actions. Every time you add a transition, SeedDMS appends it to the table on the right and also adds it to the diagram. Once a complete workflow is available, users can assign it to each document they upload into SeedDMS and then follow it throughout the whole lifecycle.

In the simplified workflow of Figure 9, the user *Enzo* is responsible for declaring the end of the *prestudy* phase for every document that must follow the *Product Introduction* workflow. Figure 10 shows how this actually looks for Enzo. When clicking on the *Workflow* tab of every document assigned to him, Enzo will see that the document still is in *prestudy*. In that case, he has the option to click on *end prestudy* to make the document progress to the *first review* phase. Also note how, unlike in Figure 9, which represents a whole workflow for all users, all phases following *first review* in Figure 10 are grayed out. The reason is that Figure 10 is the workflow for a specific document, which can only progress one phase at a time and in the defined order.

In real life, to make a team actually manage its documents using SeedDMS, you need to configure three


Workflow states management

prestudy

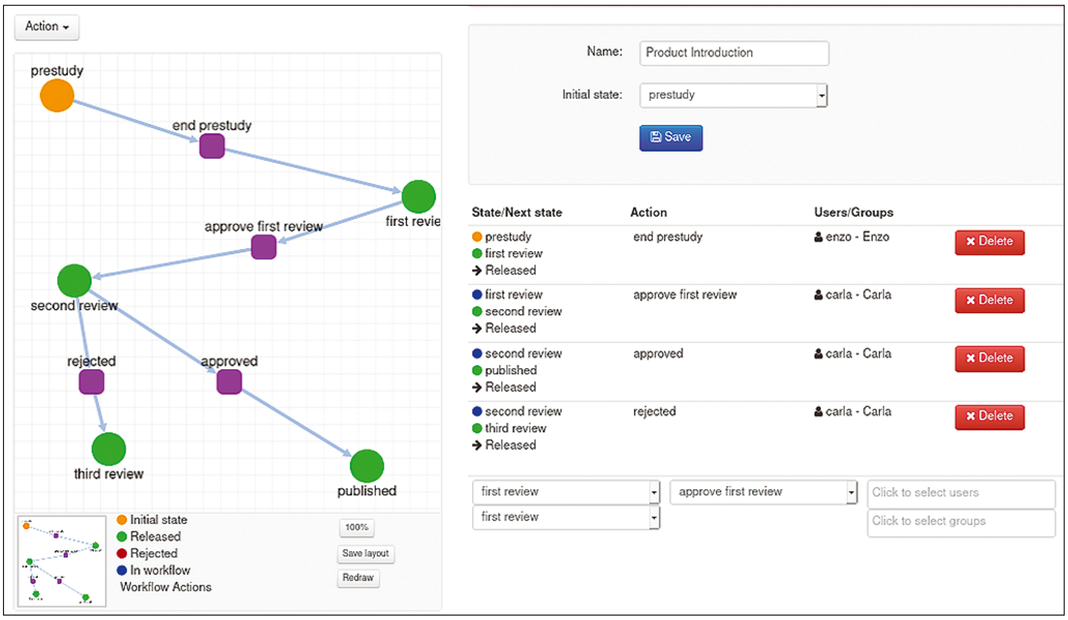
Workflow	Previous state	New state
Product Introduction	● prestudy	● first review

This state is currently used by workflows.

Name: prestudy

Document status: Released

Figure 8: Before connecting document management phases in one coherent workflow, you must assign each of them a document status.



Name: Product Introduction

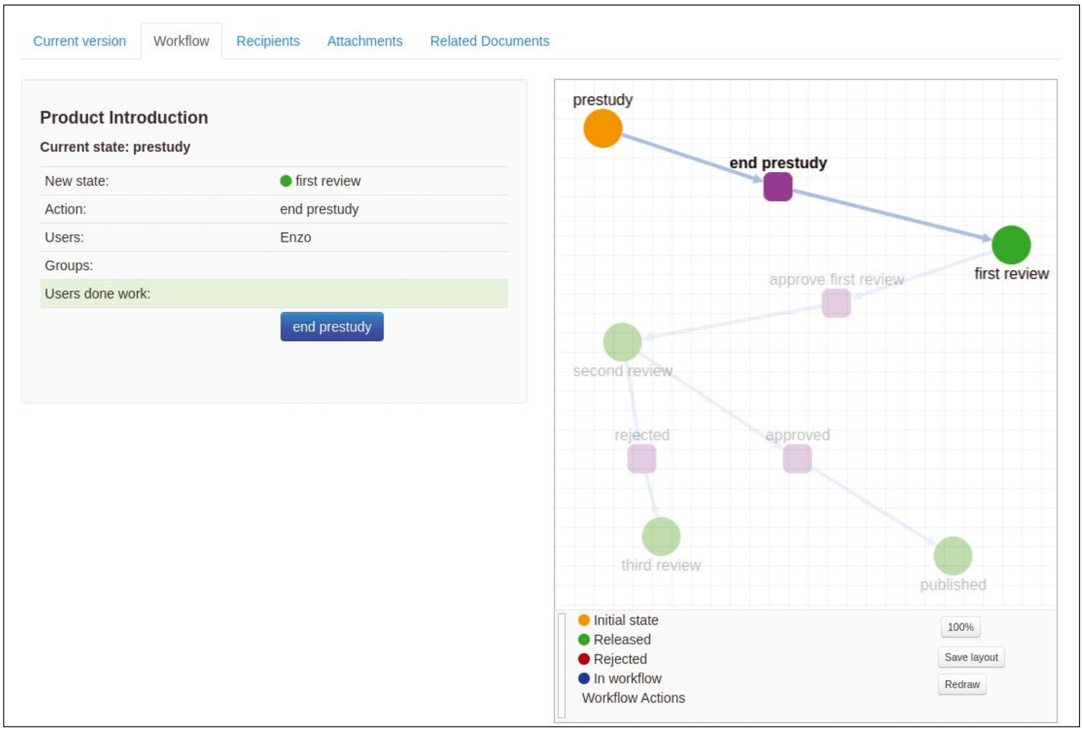
Initial state: prestudy

State/Next state	Action	Users/Groups	
● prestudy ● first review → Released	end prestudy	enzo - Enzo	✖ Delete
● first review ● second review → Released	approve first review	carla - Carla	✖ Delete
● second review ● published → Released	approved	carla - Carla	✖ Delete
● second review ● third review → Released	rejected	carla - Carla	✖ Delete

first review | approve first review | Click to select users

first review | Click to select groups

Figure 9: SeedDMS describes each workflow both in textual and graphical form.



Current version | Workflow | Recipients | Attachments | Related Documents

Product Introduction

Current state: prestudy

New state:	● first review
Action:	end prestudy
Users:	Enzo
Groups:	
Users done work:	

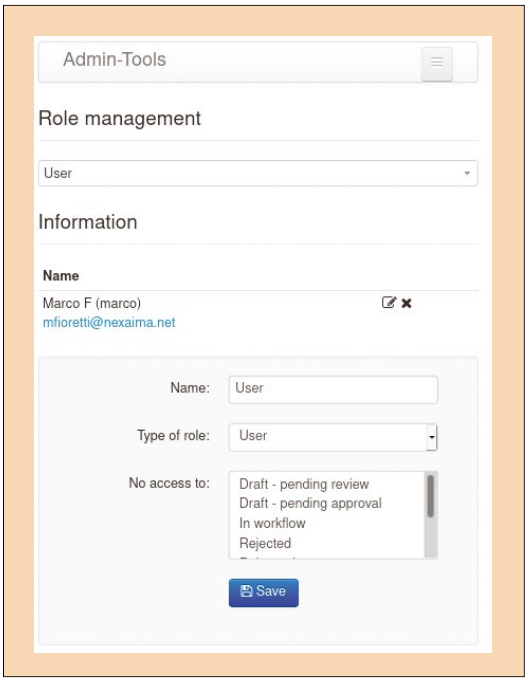
end prestudy

● Initial state
● Released
● Rejected
● In workflow
Workflow Actions

Figure 10: Workflows in action: Each user immediately sees what he or she must do on every document.

critical functions: role management, access rights, and notifications. In the roles management interface (Figure 11), you can specify which users can act on certain phases of each workflow. For instance, you can declare that only team leaders can reject a document.

The notifications feature lets users know when they must do something (refer to Figure 7). In the sample workflow of Figures 9 and 10, Carla should immediately receive an email every time Enzo ends the *prestudy* phase for a document that is also assigned to Carla, because it is her responsibility to lead the first review and eventually declare its conclusion. ■■■



Admin-Tools

Role management

User

Information

Name

Marco F (marco)
mfioretti@nexaima.net

Name: User

Type of role: User

No access to:
Draft - pending review
Draft - pending approval
In workflow
Rejected

Save

Figure 11: SeedDMS users can see only the actions and workflow phases that they have been explicitly authorized to work on.

Conclusion

Several online reviews lament the lack of complete documentation for SeedDMS. As of May 2021, I consider this critique justified. Figuring out why and how to configure every option of all the features should have been easier. The SeedDMS website does host a support forum. Several extensions are for sale at a dedicated SeedDMS marketplace [5], and if you really need customization, the main developer is available for consulting work. All in all, if your team has to work on many documents that follow strict workflows, I would say give SeedDMS a try. ■■■

The Author

Marco Fioretti (*http://mfioretti.com*) is a freelance author, trainer, and researcher based in Rome, Italy. He has been working with free/open source software since 1995 and on open digital standards since 2005. Marco also is a Board Member of the Free Knowledge Institute (*http://freeknowledge.eu*) and blogs about digital rights at *https://stop.zona-m.net*.

Info

[1] SeedDMS: *https://www.seeddms.org/*

[2] SeedDMS demo: *http://demo6.seeddms.org/*

[3] Play with Docker: *https://www.docker.com/play-with-docker*

[4] How to Create User and Database MySQL via Command Line: *https://blog.bestariwebhost.com/en/how-to-create-user-and-database-mysql-via-command-line-cli/*

[5] SeedDMS extensions marketplace: *https://shop.mmk-hagen.de/*

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.

#248/July 2021

Brain Tools

Sometimes you want the computer to think for you, and sometimes you want the computer to make you think. This month we present a selection of free Linux tools for learning and thinking.

On the DVD: Ubuntu 21.04 and Fedora 34 Workstation

#247/June 2021

Post-Quantum Encryption

Quantum computers are still at the experimental stage, but mathematicians have already discovered some quantum-based algorithms that will demolish the best of our current encryption methods. What better time to look for quantum encryption alternatives?

On the DVD: Knoppix 9.1 and ZORIN OS 15.3 Core

#246/May 2021

Faster Startup

Weary of waiting for a login window? Your driver-drenched Linux distro was configured for all systems, not for your system. This month we show you how to optimize your system for faster startup.

On the DVD: Manjaro KDE Plasma 20.2.1 and Clonezilla Live 2.7.1

#245/April 2021

Choose a Shell

You're never stuck with the same old command shell – unless you want to be. This month we review some of the leading alternatives.

On the DVD: Arch Linux 2021.02.01 and MX Linux mx-19.03

#244/March 2021

Stream Processing

The explosion of real-time data from sensors and monitoring devices is fueling new interest in alternative programming techniques. This month we wade into stream processing.

On the DVD: FreeBSD 12.2 and GhostBSD 20.11.28

#243/February 2021

iNet

With Linux, more innovation is always on the way. This month we take a look at the iNet wireless daemon, a new wireless client that is poised to replace the venerable WPA Supplicant.

On the DVD: Linux Mint 20 and Kali Linux 2020.4

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 her

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.

GUADEC 2021

Date: July 21-25, 2021

Location: Virtual Event

Website: *https://events.gnome.org/event/9/*

GUADEC is the GNOME community's largest conference, bringing together hundreds of users, contributors, community members, and enthusiastic supporters for a week of talks and workshops. View the GUADEC schedule and start planning your GUADEC experience.

USENIX Security '21

Date: August 11-13, 2021

Location: Virtual Event

Website: *https://www.usenix.org/conference/usenixsecurity21*

Please join us for the 30th USENIX Security Symposium. USENIX Security brings together researchers, practitioners, system administrators, system programmers, and others to share and explore the latest advances in the security and privacy of computer systems and networks.

Events

GUADEC 2021	July 21-25	Virtual Event	https://events.gnome.org/event/9/
USENIX Security '21	August 11-13	Virtual Event	https://www.usenix.org/conference/usenixsecurity21
Developer Week Global: Cloud	September 14-15	Virtual Event	https://www.developerweek.com/global/conference/cloud/
KVM Forum	Septermber 15-16	Virtual Event	https://events.linuxfoundation.org/kvm-forum/
Open Source Summit North America	September 27-30	Seattle, Washington	https://events.linuxfoundation.org/
Embedded Linux Conference North America	September 27-30	Seattle, Washington	https://events.linuxfoundation.org/
DrupalCon Europe 2021	October 4-7	Virtual Event	https://events.drupal.org/europe2021
JAX London Hybrid	October 4-7	London, UK and Online	https://jaxlondon.com/
IEEE Quantum Week 2021	October 18-22	Virtual Event	https://qce.quantum.ieee.org/
SeaGL (Seattle GNU/Linux Conference)	November 5-6	Virtual Event	https://seagl.org/
Linux Storage Filesystem & MM Summit	December 6-8	Palm Springs, California	https://events.linuxfoundation.org/lsfmm/
DeveloperWeek Global: Enterprise	December 7-8	Virtual Event	https://www.developerweek.com/global/conference/enterprise/

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.*

Authors

Dirk Ambras	16
Mats Tage Axelsson	76
Erik Bärwaldt	24, 50
Zack Brown	12
Bruce Byfield	6, 22, 36
Joe Casad	3
Stefano Chittaro	30
Mark Crutch	69
Marco Fioretti	88
Jon “maddog” Hall	70
Charly Kühnast	34
Rubén Llorente	40
Andrew Malcolm	60
Vincent Mealing	69
Pete Metcalfe	54
Graham Morrison	82
Tim Schürmann	71
Scott Sumner	44
Jack Wallen	8

Contact Info

Editor in Chief
Joe Casad, jcasad@linux-magazine.com

Copy Editors
Amy Pettle, Aubrey Vaughn

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
Dena Friesen

Cover Image
© georgejmclittle, Daniil Peshkov, and andre266, 123RF.com

Advertising
Brian Osborn, bosborn@linuxnewmedia.com
phone +49 8093 7679420

Marketing Communications
Gwen Clark, gclark@linuxnewmedia.com
Linux New Media USA, LLC
4840 Bob Billings Parkway, Ste 104
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.

Printed in Nuremberg, Germany by hofmann infocom GmbH.

Distributed by Seymour Distribution Ltd, United Kingdom

LINUX PRO MAGAZINE (ISSN 1752-9050) is published monthly by Linux New Media USA, LLC, 4840 Bob Billings Parkway, Ste 104, 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, 4840 Bob Billings Parkway, Ste 104, Lawrence, KS 66049, USA.

Published monthly in Europe as Linux Magazine (ISSN 1471-5678) by: Sparkhaus Media GmbH, Bialasstr. 1a, 85625 Glonn, Germany.

NEXT MONTH

Issue 250

Approximate On Sale Date

UK / Europe	Aug 07
USA / Canada	Sep 03
Australia	Oct 04

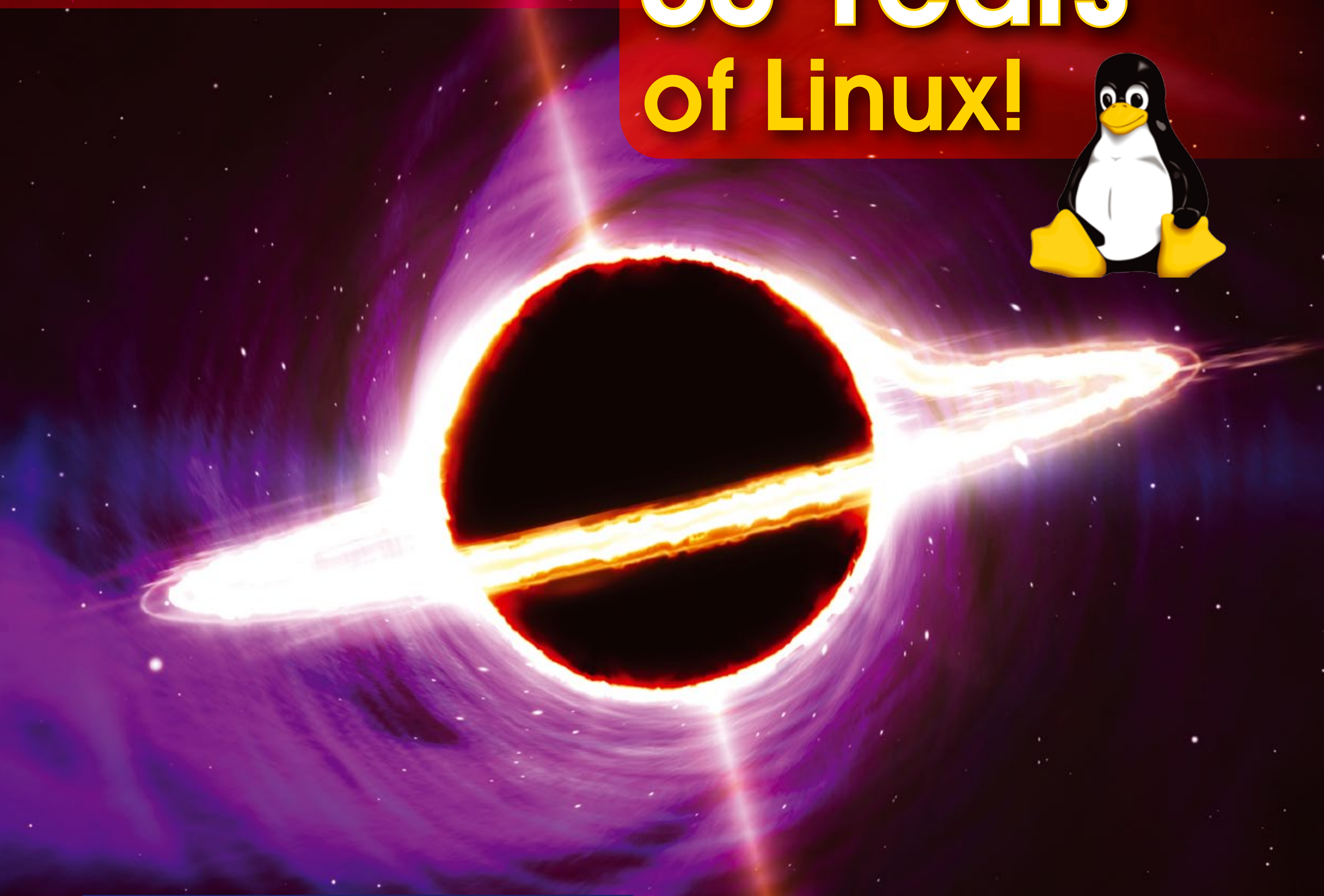
Please note: On sale dates are approximate and may be delayed because of logistical issues.

Issue 250 / September 2021

Linux Kernel

Next month we celebrate 30 years of Linux with a special look at the ever-changing Linux kernel.

30 Years of Linux!



Preview Newsletter

The Linux Magazine Preview is a monthly email newsletter that gives you a sneak peek at the next issue, including links to articles posted online.

Sign up at: ***https://bit.ly/Linux-Update***

Lead Image © skorzewiak, 123RF.com