



VIDEO EDITING

Get better videos in Ubuntu with Kdenlive



WRITE A NOVEL

Grab the best open source writing tools



GET CHATTING!

We test the most powerful Matrix clients

LINUX FORMAT

The **#1** open source mag

NEW FOR 2023

10 HOT Pi PROJECTS

Break out your Raspberry Pi and Picos to discover fun and thrilling builds for all!

PLUS!

How to contribute to open source projects

HOW TO...

- » Master GIMP layers
- » Get started with SerenityOS
- » Take better notes with Obsidian

MONITOR PACKETS

Keep an eye on your TCP/IP network traffic

EASY SIGN-UPS

Create QR codes for fast and simple logins

CODE WEB APPS

Play with our Python-based catalogue tool

LXF February 2023

FUTURE



PAPER POWER

60% of the energy used to produce paper and paper packaging in Europe comes from renewable sources.

Discover the story of paper

www.lovepaper.org

Source: Confederation of European Paper Industries (CEPI), 2018
CEPI represents 92% of European pulp and paper production





LINUX FORMAT



We're kicking off 2023 with a selection of hot Pi projects – what maker builds or new open source ventures would you recommend for readers to try this year?



Matthew Holder

The key to learning a new skill is to tackle it in bite-sized chunks. For example, if you wish to learn about programming and using GPIO, decide what your end goal is and start small. On day one, make an LED blink, on day two, read the state of a switch, and build from there.



Nick Peers

This year's hot new find for me is the fabulous Audiobookshelf, a self-hosted streaming server for audiobooks and podcasts. Since its debut in September 2021, it's come on in leaps and bounds to

become the best in its field by some way. Who needs Audible?



Les Pounder

The ESP32 range of boards have most excited me in 2022, so for 2023 I implore you to give one a go. The Banana Pi Pico W shares the same pinout as the Raspberry Pi Pico – Adafruit's range of ESP32 boards

cannot be beaten for quality and support.



Michael Reed

For a new year's resolution, I ought to get a Nextcloud server on my Pi. It would be nice to have all my file and media sharing in one place, because at the moment, I seem to be relying on about three different

proprietary solutions whenever I need to get at my stuff.



Mayank Sharma

I've been thinking of building OpenCat, a Pi-powered robotic cat, since forever. It can pose, crawl, balance, recover from falls, actually does as you say, and doesn't poop on your sofa. Funnily, this build always

seems to get on my new year wish list but doesn't come off it.

Build the future



Why didn't you pick up that flashy MacFormat or shiny T3 magazine? Perhaps you don't own a Mac, or don't care about iPhones or Samsung Folds – I hear they're very slick. Do you hate technology? Oddly, you can't actually buy a Future-published Windows magazine any more – people lost interest years ago...

Perhaps that's it – you're interested in how things work, you're interested in what goes

on underneath and what makes a system tick. While Apple, Microsoft and much of the tech world are desperate to hide all the complexity away, there are still people who enjoy getting their hands dirty and learning what makes something run.

So when Eben Upton first revealed the Raspberry Pi's naked PCB back in 2012, much of the tech world moved on for lack of a shiny alloy case to attract its attention. But makers, the open source community and the curious brought down the pre-order web servers faster than any mob could move. With sales now topping 50 million units, as much to the commercial sector as enthusiasts, the Raspberry Pi Foundation, with its primary focus on education, is perfectly positioned to help build futures – for children, for open source projects, for makers and for industries.

So, if you have a Raspberry Pi or Pi Pico and are wondering what you can do with it, why not build yourself a future? Jump to page 32 and start having fun with our selection of hot projects for 2023 – you're bound to enjoy them all!

Neil

Neil Mohr Editor
neil.mohr@futurenet.com



Subscribe
& save!

On digital and print:
see page 16

Contents



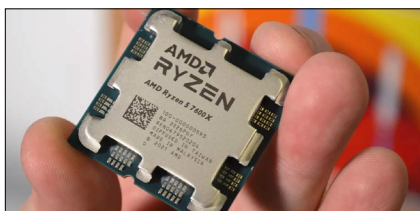
REVIEWS

Wix..... 19

A giant in the website builder industry, **Shashank Sharma** takes the feature-rich Wix for a test drive and discovers there's plenty to like under the bonnet.

AMD Ryzen 5 7600X..... 20

AMD's latest Ryzen offering is a beastly 'budget' gaming CPU but one that **Christian Guyton** has discovered has a big if temporary drawback.



WattOS R12..... 21

Nate Drake explores how WattOS can magically bring your ancient hardware back to life with this zippy, if spartan Linux distribution.

Kali Linux 2022.04..... 22

If you exist in the grey twilight between good and evil, and want to look cool while doing it, **David Rutland** has just the deity-based distro for you.

Nitrux 2.5.1..... 23

An impressive beginner distro that's well worth your time, even if you spend most of that tweaking its many, many settings, says **Mayank Sharma**.

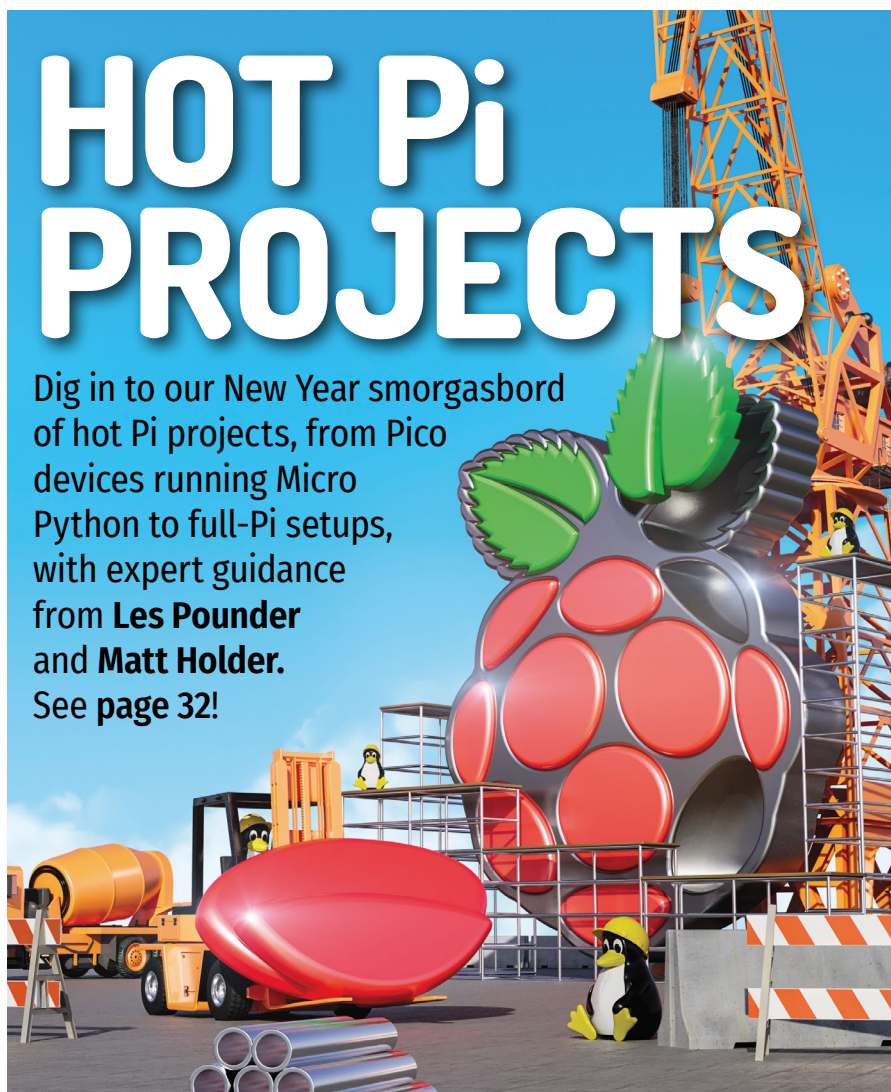
Victoria 3..... 24

"The peasants are revolting," cries The Management, but really it's just **Jonathan Bolding** serving up their afternoon tea and biscuits.

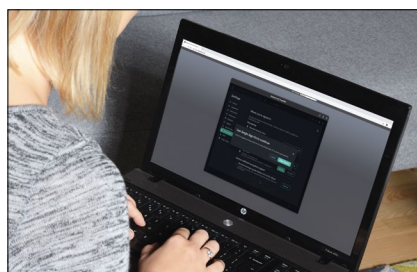


HOT Pi PROJECTS

Dig in to our New Year smorgasbord of hot Pi projects, from Pico devices running Micro Python to full-Pi setups, with expert guidance from **Les Pounder** and **Matt Holder**. See page 32!



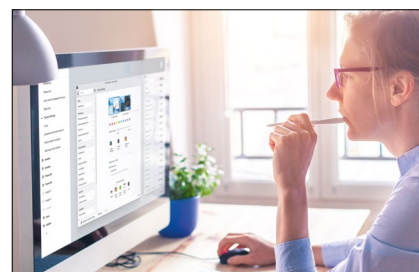
ROUNDUP



Matrix message clients..... 26

Mats Täge Axelsson examines the potential of distributed messaging and federated authentication servers that use the Matrix protocol.

IN DEPTH



Contributing for non-coders..... 44

Aaron Peters, who is a coding tinkerer at best, describes some ways that non-technical folks can contribute to open source projects.

PI USER

Pi news 49

Introduced by **Neil Mohr**, who can't get enough Pi. Christmas deliveries of 100,000 Raspberry Pis, plus two fruity Pi alternatives.

Petoi Bittle Robot Dog 50

The rise of quadruped robots is upon us — their mission to boost STEM and AI learning with **Anj Bryant**.

Elegoo Neptune 3 Pro 51

Denise Bertacchi wishes everything in life could be relied upon to add more features while remaining excellent value.

Pi robots: motors and control 52

In part two of his series, **Les Pounder** shows us how to safely connect and control motors with a Raspberry Pi.



CODING ACADEMY

Monitor TCP/IP networks with eBPF 90

Naturally suspicious, **Mihalis Tsoukalos** keeps an eye on all his network traffic using eBPF — and he'll show you how, too.

Finish your cataloguing app 94

With a tear in his eye, **Matt Holder** finishes up our cataloguing database app by adding search and export functionality.

```

[19:43:15]
size      : count  distribution
0 -> 1      : 0      |
2 -> 3      : 0      |
4 -> 7      : 0      |
8 -> 15     : 0      |
16 -> 31    : 7      | *****
32 -> 63    : 31     | *****
[19:43:16]
size      : count  distribution
0 -> 1      : 0      |
2 -> 3      : 0      |
4 -> 7      : 0      |
8 -> 15     : 0      |
16 -> 31    : 32     | *****
32 -> 63    : 94     | *****
64 -> 127   : 4      | **
128 -> 255  : 3      | **
256 -> 511  : 1      |
512 -> 1023 : 1      |
1024 -> 2047: 7      | **
[19:43:17]
size      : count  distribution
0 -> 1      : 0      |
2 -> 3      : 0      |
4 -> 7      : 0      |
8 -> 15     : 0      |

```

REGULARS AT A GLANCE

News 6

Tumbleweed moves to x86-64-v2 — will you be left behind? Rust grows in the kernel, Asahi Linux ports Linux to M1/M2 Macs, and AMD opens up ray tracing.

Kernel watch 10

Answers 11

We help you with remote desktop work, forgotten passwords, merging partitions, exploring SSH tunnelling, YouTube downloading and much more.

Mailserver 14

Linux Format lends a hand, a look back at one reader's long career in computing, and we learn about an amazing scheme repurposing ageing laptops.

Subscriptions 16

Get your monthly Linux dose and save cash!

Back issues 68

Get hold of previous *Linux Format* editions.

Overseas subscriptions 69

Get *Linux Format* shipped around the globe.

Hot Picks 83

Mayank Sharma might have figured out what to do until F1 2023 — master all this open source software he's found for us: *Cryptomator*, *PowerShell*, *OSM*, *Logseq*, *Easy Effects*, *Ear Tag*, *Gabut Download Manager*, *3D-Puzzles*, *Total Chaos*, *Metadata Cleaner* and *Hidamari*.

Next month 98

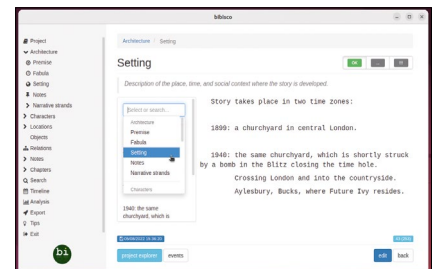
TUTORIALS

TERMINAL: Tackling tabular tables 56

With the right tools at his disposal, **Shashank Sharma** has no trouble making sense of his vast heaps of endless Formula One tabular data.

BIBISCO: Plan and write a novel 60

Nick Peers reveals how to develop better stories with the help of our favourite freemium app for budding novelists and part-time magazine authors.



KDENLIVE: Polish your video footage 64

Michael Reed looks at how you can use the best open source video editor, *Kdenlive*, to produce high-quality clips suitable for uploading all over the internet.

GIMP: Fun with layers and masks 70

Karsten Günther explains how *GIMP* manages layers, what the different types are and how they differ from channels, plus how to get the most out of them.

GIMP: Create better QR codes 74

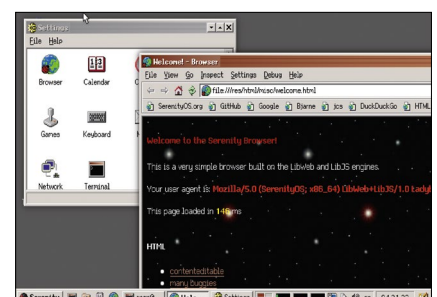
Mike Bedford reveals how to get creative with your QR codes so you can draw more attention to them and turbo-boost your newsletter sign-ups, so we're told.

OBSIDIAN: Nifty note-taking 76

Nate Drake walks you through adopting the almighty Markdown-based note-taking app to see whether it lives up to its legend and can help with his lighthouse project.

SERENITYOS: Direct from the source 78

What do you get if you take a modern kernel, a classic GUI and a handful of custom apps? **Michael Reed** has a bit of a chat with the developers behind it.



Newsdesk

THIS ISSUE: CPU support » Rust » Linux on Apple » AMD ray tracing » Thunderbird for Android » RGB lighting

DISTRO

OpenSUSE to drop early 64-bit CPUs

Tumbleweed moves to x86-64-v2 – do you even know which version you have?



Tumbleweed is SUSE's rolling release distro and is constantly updated with the latest stable software; development is handled by OpenSUSE Factory. Tumbleweed's Dominique Leuenberger has sent an email to the Factory mailing list saying the Factory is "repurposed to move forward with x86-64-v2". This leaves the first generation of 64-bit chips dead when implemented. A new repository will be created for legacy systems but will require volunteers to keep it running as, after setting it up, Leuenberger says he won't touch it. SUSE Linux Enterprise is already due to move to x86-64-v2 (it had considered moving to v3, but this was dropped as a step too far). Given that Leap is derived from Tumbleweed, it can only be a matter of time before that joins the list. It makes sense – Tumbleweed is aimed at high-performance and enterprise applications and the bar has to move at some point.

Not all 64-bit chips are the same but you will be forgiven for not knowing which generation of architecture you have. In 2020, AMD, Intel, Red Hat and SUSE got together to define the difference between various implementations of the instruction set because, over the years, numerous extensions had been added, and if you wanted to access these directly, you needed to know if a chip included support. To make this "easy", x86-64 architectures were divided into four versions, each representing a minimum level of support: x86-64-v1 is the baseline; v2 starts around 2009 and includes SSE4; v3 appeared around 2015 and includes AVX2; while

v4 has AVX512, which is limited to server chips and Intel's Core-X. Unless your system is over 10 years old, you're probably fine. The loss of earlier versions of the Atom chips might sting – some were released as recently as 2013 and were still v1 – but otherwise Athlon 64, Pentium-D and first-gen Intel Core 2 are the main victims.

Need we worry, though? Probably not. Tumbleweed isn't an average distro, nor is Leap; both target business rather than home. Red Hat Enterprise Linux 9 has announced a similar move. Support for x86-64-v1 will start to fade on bigger commercially orientated distros, but one thing the whole community is good at is building distros: whatever your hardware, if it is at all possible to run Linux on it, somebody will build a distro that can. Things have to move on at some point, and who still laments the start of the end of x86-32 support about 10 years ago? Even now there are plenty of distros that run on a 32-bit box, most notably Debian. If you are in charge of server racks, you might like to start checking your processors, though.

See the SUSE page at <https://en.opensuse.org/openSUSE:X86-64-Architecture-Levels>.

SUSE's Tumbleweed prides itself on being at the leading-edge of Linux distros. To stay there means dropping support for older 64-bit processors.



Still running an Intel Core architecture chip? Some are still decent enough!

ROLLING ON AND OVER...

Tumbleweed is aimed at high-performance applications and the bar has to move at some point

PROGRAMMING LANGUAGE

Rust testing the kernel

LLVM is ready to check your distro.

The release of Linux Kernel 6.1 introduced a significant change: support for a second language, Rust. This offers developers intriguing options. There are 30 million lines of code in the kernel, so it'll be many years before Rust becomes a significant part of it. For now, all we have is some sample modules and subsystem-specific bindings. There aren't any real device drivers yet, but the basic infrastructure has been laid down. At the last Linux Maintainers Summit, Linus Torvalds asked for this new infrastructure to be tested, and it has now been added to the KernelCI (the community-based testing project that makes sure Linux passes muster). There's a simple command to check a distro has the required dependencies, which returns the curt but friendly message "rust is available!" if they are. Currently, only version 1.62 of the LLVM-based *Rustc* compiler is supported, so there's some work to do yet. (LLVM is a set of low-level compiler tools that help develop front and back-ends for any instruction set architecture; it's language independent.) The GCC Rust front-end is currently being worked on, but for now, if you want Rust in your kernel, you need to use LLVM.

CREDIT: LLVM.org



Linux is no longer the home of pure C. If you go with Rust, kernel testing is in place, albeit in a limited form for now.

It is still very early days for Rust and Linux, but the plan is that the capability to do any required testing will keep pace, which is why basic testing has been put in place before the kernel supporting it was even released (by six days, anyway). The next step is kunit tests, more toolchain and Kconfig combinations. However enthusiastic the uptake for developing the kernel using Rust proves to be, it'll be a while before wide-scale testing of drivers and other components written in Rust is required. When Mr Torvalds asks for help with the kernel, it seems things get done pretty quickly.

APPLE

Linux and the M1

The making of Asahi Linux for Apple.

The Asahi Linux project has a grand ambition: a porting effort to get Linux running on Macs powered by Apple's own silicon, the M1/M2 chips. Work started early in 2021, not long after Apple announced it was going its own way with processors. No easy task, but it seems we'll have a functional native GPU driver sooner than we expected as we've been treated to a demonstration of a Direct Rendering Manager driver written in Rust running GNOME, along with a selection of apps including *YouTube* and *Firefox*, all on an Apple M1 chip.

The magic was attributed to one Asahi Lina, who joined the project a few months ago. Presumably this is a pseudonym, or an almost unbelievable coincidence. Now a detailed

story of how the feat was achieved has appeared in a blog by Lina entitled *Tales Of The M1 GPU* (<https://asahilinux.org/2022/11/tales-of-the-m1-gpu>), which makes for inspiring reading. It begins with Alyssa Rosenzweig reverse engineering the M1's GPU and driver, while Dougall Johnson documented the shader architecture. In no time they were drawing triangles. Enter Lina to add the kernel driver. It then gets a bit complicated and involves Python, Mesa, Rust and some coding magic. On 7th December, we got an alpha release of the first Apple GPU driver – it's buggy and hasn't passed OpenGL conformance tests, but it mostly works. At the end of this road is a fully functional Linux system running on Apple hardware, a tantalising prospect. It's a significant milestone, top work.

OPINION

TURN TO RUST



Adrian Ratiu
is a senior software engineer at Collabora.

After waiting in the Linux-next integration tree for about 18 months, the basic Rust infrastructure finally landed in the mainline Linux kernel with the release of v6.1. While this did not include any real device drivers and only a few toy sample modules, further subsystem-specific bindings with real drivers should be added in the future.

During the 2022 Linux Maintainers Summit in Dublin, Linus Torvalds asked CI systems to start testing the new Rust infrastructure. So, with that in mind, Rust testing has now been added to KernelCI!

Of course, the future of Rust in the Linux kernel in general and what community adoption Rust will have within the project remains uncertain. The codebase is still in its early stages of experimentation and some topics are controversial. However, we do plan to help test Rust code via KernelCI and this is why we've added build testing early.

Going forward, we plan to add kunit tests, more toolchain and Kconfig combinations, GCC-Rust when it eventually becomes ready, extend testing to cover more subsystems and architectures (as they are added), and just maybe, if the stars align, even do boot and runtime testing, bisections and the like for drivers implemented in Rust.

OPINION

EXPERT
ADVICE

Keith Edmonds
is MD of Tiger Computing,
which provides support for
businesses using Linux.

“Linux isn't like Windows or Mac OS. You know that, of course, but it could be news to some of your users.

What they see is a computer. Desktop, laptop, tablet; Dell, Apple, Lenovo; Android, Windows, Mac OS. It's a computer, and you fix computers, right?

Yes, you can probably make an attempt at fixing systems that you don't normally work on. There are enough similarities that you can probably help at least with the more straightforward things.

But are you an IT generalist or specialist? The generalist can be likened to your GP – they have a wide range of medical knowledge, but for most areas, they don't possess the required in-depth knowledge.

The person to be in the IT world is the specialist, not the generalist. You want to be known for providing excellent Linux support to your customers – and you don't need to worry about other systems.

The specialist has the knowledge and experience. They know when to seek more in-depth information and how to use it. You don't have to be a generalist. When you're asked to look at a system you don't know, it's OK to tell the owner that you're a Linux expert only.”

GRAPHICS

AMD opens
up ray tracing

The Radeon Raytracing Analyzer finally goes full open source.

Last summer, AMD added its Radeon Raytracer Analyzer (RRA) to its development suite to help ray tracing with DirectX and Vulkan APIs. The initial release was binary-only, although some documentation mentions the MIT Licence. Now it is properly open source as part of AMD's *GPUOpen* project, its middleware suite aimed at developers. RRA aims to ease the transition from rasterisation to ray tracing with tools to help organise model geometries and to understand and finesse ray tracing performance. Using RRA, you can visualise bounding box hierarchies and easily see areas



You now have 69,420 odd lines of AMD code to help develop ray-tracing apps on Radeon GPUs as AMD embraces open source.

that will produce a performance bottleneck. The package even includes build instructions, currently tailored to Ubuntu. Good stuff for developers as you can delve deeper into the ray tracing guts of Radeon GPUs, including the new RDNA 3. AMD's approach is laudable, unlike Nvidia's. It has a similar software suite, *Nvidia GameWorks*, which is only partially open source. Linux gaming is growing and to fully support that, you've got to go open source; it comes with the territory. Nvidia's recalcitrance looks counterproductive.

You can find *GPUOpen* at <https://github.com/GPUOpen-Drivers/libamdrdf>.

CREDIT: AMD

EMAIL

The K-9 and
Thunderbird
merger

New milestone in project to completely rebrand K-9.

The *Thunderbird for Android* project is the reworking of *K-9 Mail* until its look and feel match *Thunderbird*, at which point it'll get rebranded. The team is busy adding features – the latest is a redesign of the message view. The UI changes include a reply action with overflow menu, colour-coded account names, and organisational labels. Tap the top section of the message and it opens a bottom sheet with message details, including each recipient's contact details. This brings *K-9* into line with *Thunderbird*, but adds screen clutter that doesn't endear itself to those who prefer *K-9*'s simpler approach. The project is still a work in progress and the exact feature set isn't nailed down, so your feedback now could count. The final transformation of *K-9 Mail* is due summer 2023, when it will be fully subsumed into the *Thunderbird* family. Read the blog at <https://thunderbird.topicbox.com/groups/android-planning>.

LIGHTING

OpenRGB gets
a big update

Extensive new device support.

OpenRGB aims to cut through the bloated proprietary systems and offer a single open source way to control RGB lighting with one app. As you might expect, this means delving into a lot of inadequately documented material from myriad, often uncooperative, sources. *OpenRGB* has released a new stable version, 0.8, which has been over a year in the making. It's a major affair with 22 pages of merged and closed MRs. Along with new features, bug fixes and so forth, there's an extensive list of new device support, although there are still the inevitable omissions.



Fancy RGB lighting looks good, but each has its own control software. Enter *OpenRGB*: one lightweight app to rule them all.

CREDIT: OpenRGB.org

Distro watch

Fresh from the farm.

NOMADBSD 131R

This is a distro designed to be run from a USB flash drive and is based around FreeBSD, this time version 13.1. New features include two images, one using ZFS and one using UFS for the filesystem, with the latter having soft updates journaling. The automated graphics driver has been updated, including support for VIA, and there's a new display manager: SLiM. To keep the image size down, *LibreOffice* and some multimedia applications have been removed.



NomadBSD

This distro likes to travel on a memory stick. With automated hardware detection and setup, it's ideal for data recovery.

LINUX MINT 21.1 BETA

Linux Mint has been designed to give the classic desktop experience with custom tools to make for a convenient out-of-the-box web-based installation, ideal for beginners. There are custom menus and desktop, and many unique configuration tools. It's based on Ubuntu and is also compatible with Ubuntu repositories. This new development snapshot includes Flatpak support integrated into the Software Manager, which makes it easy to distinguish between Flatpaks and system packages, and switch between them.



linuxmint

Aimed at those looking for a simple desktop-focused intro to Linux, it has a fuss-free install and a suite of major apps.

NIXOS 22.11

Originally a research project, this independently developed GNU/Linux distro aims to be the state of the art in system configuration management. The entire OS, kernel, apps and system packages are built by the Nix package manager and stored separately from each other. It has reliable rollbacks and a reproducible system configuration. Gnome, KDE Plasma, Cinnamon and other elements have been updated. This release will also check to make sure your passwords are strong enough.



NixOS

To make a neat arrangement of packages, there are no `/bin`, `/sbin`, `/lib` or `/usr` directories; everything lives in `/nix/store`.

OPENINDIANA 2022.10

A continuation of the OpenSolaris OS, OpenIndiana offers a full open source alternative to Solaris 11. This is the first update for a year and contains updates of many components, including *Firefox*, *Thunderbird*, *LibreOffice*, *Perl*, *Mate* desktop, *Python* and more. Other upgrades include the ability to mount and install media using NFS, the latest Intel and AMD microcode updates, enhancements to BHyVe and the internal SMB server, and updated Nvidia drivers.



openindiana

Born out of the uncertainty after Oracle acquired Sun, this has grown into a full community-based open source alternative.

OPINION

HAPPY NEW YEAR



Donnie Berkholz is senior vice president, product, at Percona.

“The start of a year always brings a raft of predictions about what will be different in the next 12 months. I prefer new year's resolutions, as these are actions you can take yourself.

My first resolution is to look at platform engineering. This takes the self-service, automated approach that is common in the cloud and applies it to self-managed software, like open source projects. It brings what people like about the cloud – ease of getting started, ability to work more efficiently – and keeps the level of freedom and control in your hands that open source provides. Adopting some of these approaches should help you get more done.

Secondly, look at data ownership. Are you able to manage where your data is processed and stored? In the US, the California Privacy Rights Act will come into force in July 2023, while the EU continues to build new legislation around data. All these rule changes will affect how you design and build apps and data. Open source projects have made it easier to manage this kind of requirement, but check if you need to make any changes.

Lastly, find a way to support your local community. If you can give back around documentation, code or financial support, please do so. It makes a big difference.”

OPINION

BYE, 2022!



Jon Masters is a kernel hacker who's been involved with Linux for over 22 years, and works on energy-efficient Arm servers.

“I’m writing this as we near the end-of-year holiday period, a time when we tend to look back at the past 12 months. And what a year 2022 was. My wife and I bought our first house – and five Linux kernels were released! We started the year with Linux 5.16, which added memory folios (improving the scaling of certain memory operations) and support for Intel AMX. Next came Linux 5.17, which added a P-state (power) driver for AMD systems, and Linux 5.18, with scheduler improvements and a switch to the C11 standard.

The summer brought Linux 5.19, which added support for AMD’s SEV-SNP and Intel’s TDX Confidential Computing extensions. These allow for the creation of specially encrypted virtual machine guests that can’t be spied upon by the hypervisor. Autumn brought Linux 6.0, which includes Runtime Verification (RV), which aims to provide some reassurance for use in more safety-critical environments (such as in cars).

And we round the year off with Linux 6.1 (see right). A tremendous amount was achieved during the third year of a global pandemic in which many of us were also trying to catch up on lost time seeing friends and family, or in some cases confronting the horrors of war. I hope you have a more relaxing 2023.”

Kernel Watch

Jon Masters keeps an eye out for all the latest happenings in the Linux kernel, so you don’t have to.

Linus Torvalds announced the release of Linux 6.1 just in time for the holidays. On the one hand, users could see it as a nice Christmas present, but it made the whole end of year period quite frantic for kernel developers. The reason for this is that following a new kernel release typically comes a two-week merge window, during which disruptive changes planned for the next release are merged into the code base (these are subsequently stabilised in the RC or Release Candidate kernels).

As we’ve mentioned before, Linux 6.1 is likely to be the next Long Term Stable kernel maintained by Greg K-H (Kroah-Hartman) for several years through stable backports. We’ve also mentioned that it is the first kernel to include support for the Rust programming language. While these are interesting headline features, perhaps even more interesting is the number of internal changes in 6.1. These include a new data structure – the maple tree – to replace red-black trees for tracking virtual memory areas (VMAs), multi-generational LRU patches that impact swapping, and various changes to improve memory-tiering support.

Maple trees replace red-black trees, a data structure optimised for frequent insertion and deletion, as well as good search characteristics. They are a good (yet

complex) data structure with a long history, and commonly used in Linux for VMA management. VMAs are how regions of virtual memory are maintained in the kernel. One problem with “rbtrees” is that “walking the nodes in numerical order is not very efficient”. Maple trees aim to improve upon this and scale up to much larger systems with less overhead.

Multi-generational LRU aims to improve the process by which Linux decides what data in memory should be paged (swapped) out to disk. Applications commonly allocate more virtual memory than they either use or is available, with the kernel swapping unused data out to a swap partition on disk. CPU hardware known as an MMU (memory management unit) helps swap it back in again when needed, but this can have tremendous overhead if in a critical path.

Finally, Linux 6.1 gains enhancements for memory-tiering. This is where machines don’t just have the traditional NUMA (non-uniform memory access) nodes associated with RAM chips that are closer to CPU sockets but also have different categories of memory. For example, with CXL (Compute Express Link), a machine might have DDR memory attached to the CPU, and also in a separate box connected using a CXL link. That separate memory might be large, but slower to access than local RAM. Tiering helps to manage these differences by deciding where to place data between chips.

» ONGOING DEVELOPMENT

Right on cue, following the Linux 6.1 announcement, Regzbot (the regression robot maintained by Thorsten Leemhuis) sent a notice of the 44 known regressions in Linux 6.1, including a few new crash bugs.

These are likely to be addressed in the stable kernel updates from Greg K-H. Khalid Aziz posted patches adding “support for sharing page tables across processes (previously mshare)”. Sharing of memory regions between two processes is far from a novel concept – it’s used for dynamic shared libraries – but traditionally, this still involved some duplication of the data structures used to track the actual memory that was shared. With these patches, the

data structures (known as page table entries) are actually shared as well. This reduces memory overhead, particularly on large systems with lots of sharing.

SeongJae Park (Amazon) leads the Linux DAMON project, an attempt to improve the memory management mechanisms for modern workloads, particularly given that “DRAM has not been increased enough to accommodate such huge working sets”. While the project is really interesting, it is also charming to see SeongJae organising a regular “DAMON beer/coffee/tea chat series” for interested folks to meet virtually. It is something that, surprisingly, doesn’t happen often for particular kernel teams.

Answers

Got a burning question about open source or the kernel? Whatever your level, email it to answers@linuxformat.com



Neil Bothwick
eats Linux
desktop
problems for
breakfast!

Q Two PCs, one desktop

I am using a CentOS 8 Stream desktop at work to run some software that uses a GUI. I want to be able to connect to it remotely from a Windows machine to continue working on a project from home or simply from another Windows PC in the office. So far, every remote desktop solution I tried either didn't work at all or only opens a new blank session in Gnome. All I want is for it to work as the Remote Desktop works in Windows, where I can always connect to my Windows work PC from home and I will see the screen exactly the way I left it before leaving for home.

Victoria Martin

A The usual remote desktop systems, such as VNC and X2Go, work by creating a new session when you connect. There is a way to do what you need with the `X11vnc` program. This should be in your distro's repositories, so install it in the usual way. Then run it from a terminal on the computer you want to control:

```
$ x11vnc -display :0
```

Run this as a normal user, and the value for `-display` is the number of your display, which is usually `:0` if you are running a

single X session. You must have a desktop session already open – it doesn't work if your computer is displaying the login screen. The program outputs a stack of information; near the end you should find something like

```
The VNC desktop is: remote-host:0
PORT=5900
```

The port number can vary – VNC starts from 5900. The number after the hostname is the port number less 5900. This is what you pass to your VNC client on the local computer:

```
$ vncviewer remote-host:0
```

We used `Vncviewer` in this example, but any VNC client works. You don't even need to remember to run `X11vnc` before you leave for home if you have SSH access – you can run it over an SSH connection. That is why we pass the display name to `X11vnc` explicitly rather than using `$DISPLAY`, as that variable is not set for an SSH session. Unlike a traditional VNC server, `X11vnc` only runs for one connection; when you exit the client, the server quits, too.

One thing to be aware of is that you are only using SSH to start the server – your VNC session is not encrypted. This may be an issue, particularly as you said you are

using it for work, but you can protect your data with an SSH tunnel:

```
$ ssh -t -L 5900:localhost:5900 remote-
host 'x11vnc -localhost -display :0'
$ vncviewer localhost:0
```

The first line sets up the SSH tunnel to `remote-host` and starts up `X11vnc`, then you run `Vncviewer` for `localhost` to go through the tunnel. If the first command shows a port other than 5900, you can press `Ctrl+C` to exit then run the command again with the port that was indicated.

Q More users than passwords

I inadvertently clicked on an icon that changed access to a second user login. I was unaware there was a second user with my name – it may have been there from when I loaded Mint 21 less than six months ago. I don't recall the password so I'm searching under all rocks and trying old passwords to no avail. Now I can't get into my system. Is there a way to change the login back to the previous setting or otherwise gain access?

Kieran Thompson

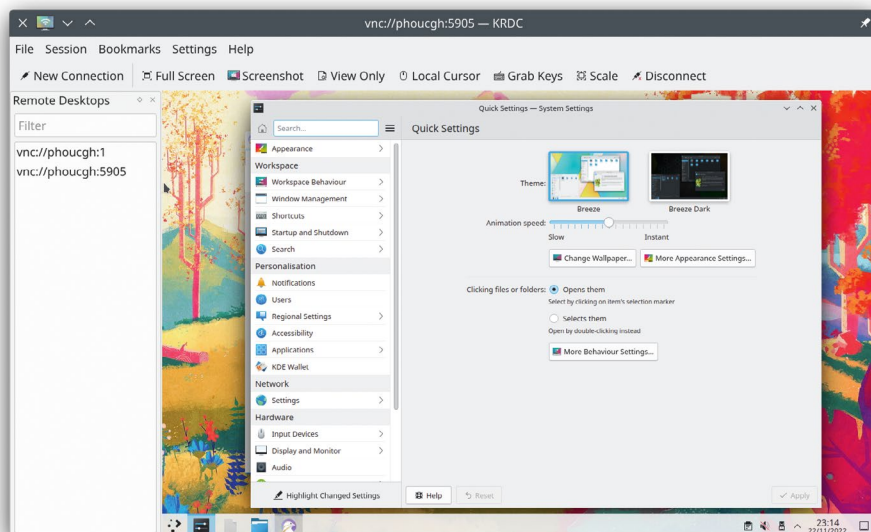
A The login screen shows the default or last selected user; you can still click on the username field to switch to the other user and log in as before. If this is not possible for any reason, you can boot from a live distro to change your user's password. Any live distro will do – I always keep a copy of `SystemRescue` (www.system-rescue.org) handy, but you can use the USB stick from which you installed Mint. Once you have booted the live distro, open a terminal and run:

```
$ sudo -i
```

This gives you root access for the rest of the session and saves having to use `sudo` with every command. Now you need to find out where your Mint system is installed. Running `$ fdisk -l` lists all your disks and partitions. Hopefully, you only have one partition marked as **Linux filesystem**, otherwise more detective work is needed. Once you know where the system is installed, mount it in the terminal:

```
$ mkdir -p /mnt/mint
```

```
$ mount /dev/sda1 /mnt/mint
```



■ The `X11vnc` program enables you to make an existing desktop session available over VNC.

Replace `sda1` with the partition where your system is actually installed. The `passwd` command is used to manipulate user passwords. It normally operates on the running system but can be instructed to work with the files of another system:

```
$ passwd --root /mnt/mint youruser
```

Here, `youruser` is the username you wish to change. You are prompted for the password twice. Unmount the Mint partition with `$ umount /mnt/mint` and reboot. Your new password should now be effective.

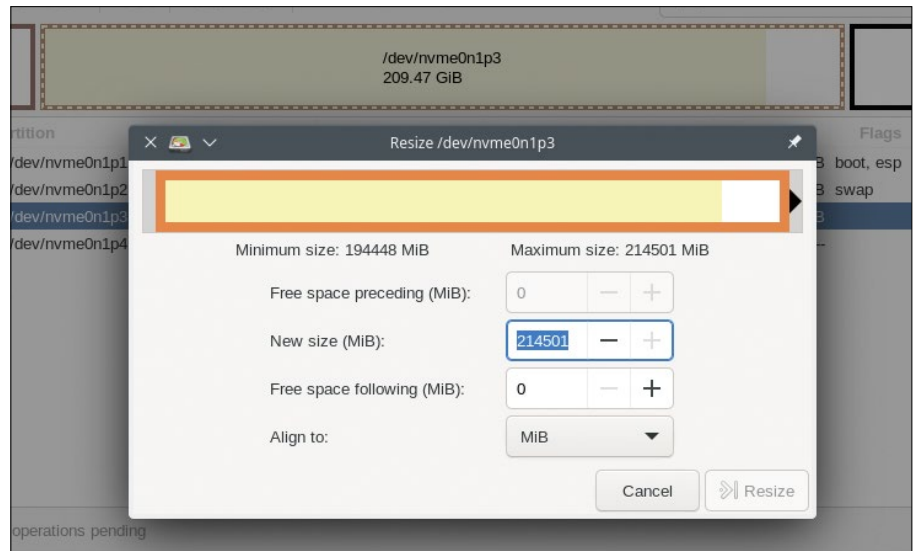
If you are unsure about the correct username, look at `/mnt/mint/etc/passwd`. The `/etc/passwd` file contains details of all the users existing on a system (although not their passwords, despite the name). There are a lot, most used by system daemons; any actual users appear near the end and have a user ID (the first number after the name) of at least 1000.

Q Merging partitions

How do I merge two partitions? One is the system and the other contains disposable content, and is directly after the system one. I'd prefer to use `Gdisk` unless you have a better suggestion.

Patrick Giles

A As a terminal junkie, I would still say the easiest way to do this is to use `Gparted` (<https://gparted.org>). Simply delete the second partition, resize the first to fill the space and hit Apply. This takes care of resizing both the partition and the



While `Gdisk` and `Cgdisk` are good for manipulating partitions in a terminal, you still have to resize filesystems separately. `Gparted` does it all in one go.

filesystem on it, but it is also simple to do with command-line tools like `Gdisk`. There is also `Cgdisk`, an ncurses version of `Gdisk`. This lets you see more clearly what you are doing. Whichever you use, the process is the same. First, delete both partitions. That sounds scary but it is fine, unless you do anything daft like writing the new partition table and rebooting before doing anything else. Next, create a new partition using all the space previously occupied by the two partitions. Set its type to Linux, then save the partition table.

Because the new partition starts at the same point as the first of the

old partitions, the filesystem is still there. However, it is still the same size as before – you have only resized the partition, not told the filesystem to use all of the space. How you do this depends on the filesystem in use, but if it is an ext4 filesystem, the Linux default, you would use the `resize2fs` command:

```
$ sudo resize2fs /dev/sdxN
```

Replace `sdxN` with the address of your new partition. If you do not give a size option, it expands the filesystem to fill the partition, and growing filesystems like this can be done while it is mounted and in use. Having said that, it is a good idea to carry out the whole process from a live distro if possible. Executing these commands from a running system should be safe, but if anything goes wrong you need to boot from a live distro to fix it, so you might as well have one available at the start.

Q Capturing Cron

I have a `Cron` job running a script every hour. I want it to email me the output and write it to a log file – `stdout` and `stderr`. This is what I have in `crontab`.

```
MAILTO="me@myemail.com"
DATEVAR="date -d '+1 hour'
+%\%d_\%m_\%Y_\%H"
0 *** root /root/reminders/tasks/hourly
>> /root/reminders/logs/$DATEVAR
```

I receive the email, but it only contains any errors that the script outputs. A log file is created, containing the output of the script, minus the errors that I received in the email (`stdout`), but the name of the log file is not the date/time – it's the literal value of the variable (`date -d '+1 hour' +%\%d_\%m_\%Y_\%H`)

I am using Debian 11.5 and believed that `Vixie Cron` could handle variables.

Hayden Hartley

» A QUICK REFERENCE TO... SSH TUNNELLING

An SSH tunnel is an SSH connection between two computers that is used to transfer data securely between the two. For example, `ssh -L 8080:localhost:80 example.com` results in all connections to port 8080 on the local computer being forwarded to port 80 on `example.com`. There are various reasons you may want to do this. There may be a restriction that prevents you directly connecting to port 80 on the remote computer with a browser. Or you may wish to send sensitive data with a program that does not provide its own encryption. All traffic going through the tunnel is encrypted by SSH. A good example is the VNC problem answered on the previous page – VNC is not encrypted but you may not want anyone to be able to snoop on what you are doing on your desktop.

The above example uses the `-L` option to set up local port forwarding,

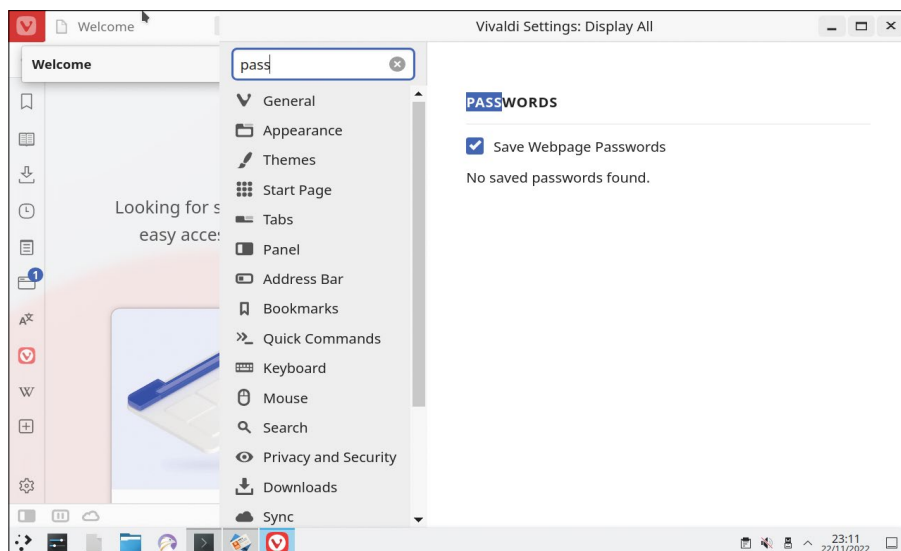
from a port on the local computer to one on the remote system. You can also set up a reverse tunnel to facilitate a connection in the opposite direction, with the `-R` option:

```
$ ssh -R 8080:localhost:80 example.com
```

In this case, we are connecting port 8080 on the remote system to the local port 80.

Although we have set up a tunnel through `localhost`, the tunnel allows connections from any computer. Another computer on your LAN can connect to port 8080 on your computer in the first example to access the remote system. You can prevent that by adding a bind address – to only allow the tunnel to be used by the system that is running it, you could do the following:

```
$ ssh -L 127.0.0.1:8080:localhost:80 example.com
```



Vivaldi's password manager enables you to export and import passwords, but be careful where you put the exported data.

A You have set **DATEVAR** to a literal string. I suspect you intended to use backticks or, preferably, a `$()` enclosure: `DATEVAR=$(date -d '+1 hour' +'%d_%m_%Y_%H')`

However, there are still issues with this as *Cron* runs in a very limited environment, so date may not even be in its path. Also, you can't depend on *Cron* evaluating the command substitution here; it is only meant for setting variables to specific values, as you have done with **MAILTO**.

The reason you are only seeing **stderr** in the mail is that all that's output as **stdout** is directed to the log file. If you remove the redirection, you should see the output in the mail, but you lose the log file. I would move the log file reporting into the script. If you still want the output to go into the mail, you can either use the `tee` command or `cat` the log file after writing to it. `somecommand 2>&1 | tee -a /root/reminders/logs/$(date -d '+1 hour' +'%d_%m_%Y_%H')`

The `2>&1` redirects **stderr** to **stdout**, then `tee` sends all its input to the given file and repeats it on **stdout**. The `-a` option appends to the file if it already exists, like the `>>` redirection you're using. Otherwise the file is overwritten, as with `>`.

If the script contains multiple commands that produce the output you need, or you'd prefer to not modify the script as you may be using it elsewhere, use a wrapper script to call it from *Cron*:

```
#!/bin/bash
/root/reminders/tasks/hourly 2>&1 | tee -a
>>/root/reminders/logs/$(date -d '+1
hour' +'%d_%m_%Y_%H')
```

Q Password transfer

Moving to a new computer, I can't move over the passwords available to

my *Vivaldi* browser. I can't find them in *Vivaldi*'s settings and I can't find them in the Gnome keyring with *Gnome Passwords and Keys* (or *Seahorse*). Neither are they to be found with *GPG*, so where are they? I am using *Ubuntu 22.04*.
Joshua Wade

A You can export and import passwords in *Vivaldi*. On the old computer, open the settings page and go to Privacy And Security. Scroll down to Passwords, or type `pass` in the search box, and you'll see an Export Passwords button. This saves your passwords as a CSV file. Importing passwords is disabled by default. On the new computer, type `vivaldi://flags` in the address bar and put `password-import` in the search box. Select Enabled and restart *Vivaldi* to apply the change. Now type `chrome://settings/passwords` in the address bar and click the menu button to the right of Saved Passwords. You will see an import option to pull in your saved passwords.

The CSV file contains all your passwords in plain text, so be careful where you store it. I would save it to a USB stick, plug that into the new computer for import, then securely wipe the USB stick, either by using `shred` to remove the file or just wipe the whole stick by using `dd` to write from `/dev/urandom` to it.

Vivaldi's data is stored in `~/.config/vivaldi` but *Vivaldi* does not state where or how the password data is kept, and being closed source, finding that out is not easy.

Q YouTube downloading

I'm on Debian 9 Stable. I've been using *YouTube-dl* for a while, but it no longer works for me. My old routine was to open lots of YouTube links in *QuiteRSS*,

then these are opened in *Firefox*. I copy all the URLs (via an add-on) and paste into a text file, then download with *YouTube-dl*. Is there an alternative?

Ben Mann

A There is a fork of *YouTube-dl* called *Yt-dlp*. The latest version of *YouTube-dl* may not be available for your distro. It can also have speed issues due to changes made at YouTube. The fork avoids such problems and adds features.

Because of the extra features, the command-line options have changed, but there is a compatibility mode to have *Yt-dlp* respond to *YouTube-dl* syntax, and even a handy wrapper script called **youtube-dl**. This means you can install *Yt-dlp* and use it as you did with *YouTube-dl*.

There is an easier way to generate the list of links. Using a command-line RSS reader such as *Rsstail* (www.vanheusden.com/rsstail) makes life a lot easier. For each feed, simply run:

```
$ rsstail -l -1 -u https://feed.url | awk
/'^Link:/' {print $2}'
```

Normally you leave *Rsstail* running and it outputs new entries every 15 minutes; the `-l` option has it run once and exit. The `-l` switch tells it to output the link for each news item. The title is also output, so we pipe the output through *Awk* to pick out only the lines starting with **Link:** and print the following URL. You can direct this to a file for later passing to *Yt-dlp*, or you can pipe the list directly to *Yt-dlp*.

Bear in mind that programs like this should not be used to circumvent copyright or fair use restrictions. **LXF**

GET HELP NOW!

We'd love to try to answer any questions you send to answers@linuxformat.com, no matter what the level. We've all been stuck before, so don't be shy. However, we're only human (although many suspect Neil is a robot), so it's important that you include as much information as you can. If something works on one distro but not another, tell us. And if you get an error message, please tell us the exact message and precisely what you did to invoke it.

If you have, or suspect, a hardware problem, let us know about the hardware. Consider installing *Hardinfo* or *Lshw*. These programs list the hardware on your machine, so send us their output. If you're unwilling, or unable, to install these, run the following commands in a root terminal and send us the **system.txt** file, too.

```
uname -a > system.txt
lspci >> system.txt
lspci -vv >> system.txt
```

Mailserver

WRITE TO US

Do you have a burning Linux-related issue that you want to discuss? Write to us at *Linux Format*, Future Publishing, Quay House, The Ambury, Bath, BA1 1UA or email letters@linuxformat.com.

You helped!

I watched a lot of YouTube videos about different distros, agonised over what OS to install... I couldn't figure out how to do it. Then I spotted your mag with Pop!_OS on its DVD. Best tenner I remember spending.

Mick

Neil says...

Thanks for the kinds words. Look everyone, we're helping, too!

Mainframe man

I've been receiving the magazine since some time in the early noughties I think, my first exposure to Linux being while I was doing a part-time MSc at De Montfort University in Leicester, 1999-2001.

We had coursework that needed to be done on Unix workstations and, as a part-time student, couldn't get access to the workstations that easily, so found Linux and started running it on a home PC.

I loved the article on retro emulation (LXF295). I was expecting something along the lines of similar articles on the likes of Apple II, Tandy TRS and so on, but it was ICL 1900 series mainframes.

My first computer was not a personal computer at all, but a mainframe: an ICL 1904S, so I absolutely loved the article on how to emulate dear old George 3. Coding sheets, card decks and the Westinghouse teletype – though we did also have a couple of MOP terminals as well. I was rewriting PLAN (ICL

assembler) programs in COBOL as part of a project for HM Treasury to replace the ageing 1900 series with the 2900 series. This was back in 1980.

I moved on from the ICL 1900 to the ICL 2946, then the NCR V series mainframe, ending up working on the IBM AS/400 around 1988.

Fast-forward a few decades and I'm still working on the IBM AS/400 successor – I've lost track of all the marketing changes. I think it's now a Power System I. The best thing about the latest incarnation: it can run Python, though most of the programming is still RPG.

Alongside the AS/400s, we introduced Linux to the business I worked for around 2002, running Apache and Tomcat with MySQL as the database. We had a bit of opposition at the time (from the US) on why we ran Linux rather than Solaris or AIX, but a few years on from that and almost all the proprietary Unix flavours had been replaced by Linux.

My laptop is still Windows (work-owned) but there is the Subsystem for Linux, and I have treated myself to a Pi 400!

Andrew Bennison

Neil says...

You've certainly kept busy over the years. Good to hear the mainframe article brought back some memories and it's interesting to hear about your transition from Unix to Linux – I bet the sales people weren't happy about you shifting to Linux! Enjoy the Pi 400 – it's probably a lot faster than those old systems and much easier to carry!

Pi gaming

Just wondering if you have any advice on open source programmable Game Boy-like devices? I remember these used to come up on occasion. My son and I want to have a go at making our own games to play.

Luke and Max, Australia

Neil says...

It really depends on how authentic and hands-on you want to get. See the project on page 34 for an outline of how to install RetroPie, which is largely the basis of all these kits, software-wise.

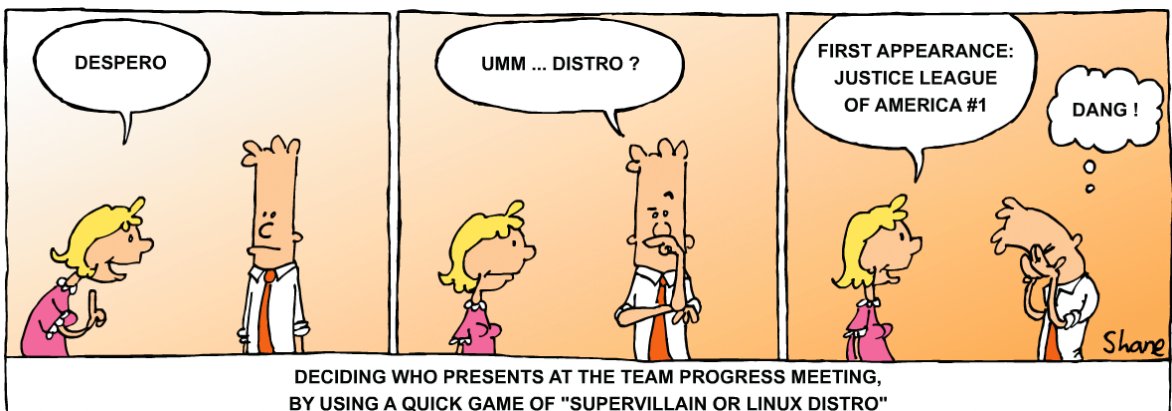
For premade handheld console kits, you're spoiled for choice; we looked at the Pimoroni Picade in LXF295. WaveShare has a range of options – we like the Game

CREDIT: UKRI Science and Technology Facilities Council.



The *Linux Format* office, circa 2021.

Helpdex





■ The Lyra+ looks like a mini Steam Deck.

HAT <https://bit.ly/lxf298pigame>. The PiBoy is also a popular range, though we've not tried it – see <https://experimentalpi.com> for more details. Similarly, the Lyra+ is a DIY Compute 4 kit from www.creoqode.com that seems very slick. Let us know how you get on!

Watching the watchers

I was quite intrigued by the content of your CCTV article (LXF293) and thought it might be interesting to at least play with a basic setup. So, my first thought was: what would a suitable camera cost? I was pretty overwhelmed by the range of options, just for cameras for external use, with PoE but no tilt and zoom. Many seem to include object recognition and other features referred to in *Frigate*. Would these clash or does that indicate the cameras are over-specified for the role? If Matt could give some kind of guide to selecting suitable cameras, that would be very useful.

John Mangan

Matt says...

I don't stick to any particular brand, but look through the listings for either RTSP or ONVIF2 support. RTSP support means it's pretty likely to be supported in something like *Frigate*, while ONVIF means it is compatible with a lot of the fully blown NVR tools. I use a Unifi camera at home and that's RTSP. **LXF**



■ You can use any camera for home CCTV, but we have some pointers...

» LETTER OF THE MONTH

Laptop drive

Over the last few months, the Isle of Wight Linux User Group (www.iwlug.org.uk) has supplied over 80 laptops to guest Ukrainian refugees. It is a joy to equip laptops with 128GB SSDs and bring them back to a new lease of life. We use Linux Mint Cinnamon by default, enabling Ukrainian and Russian languages, and stick on Cyrillic key tops.

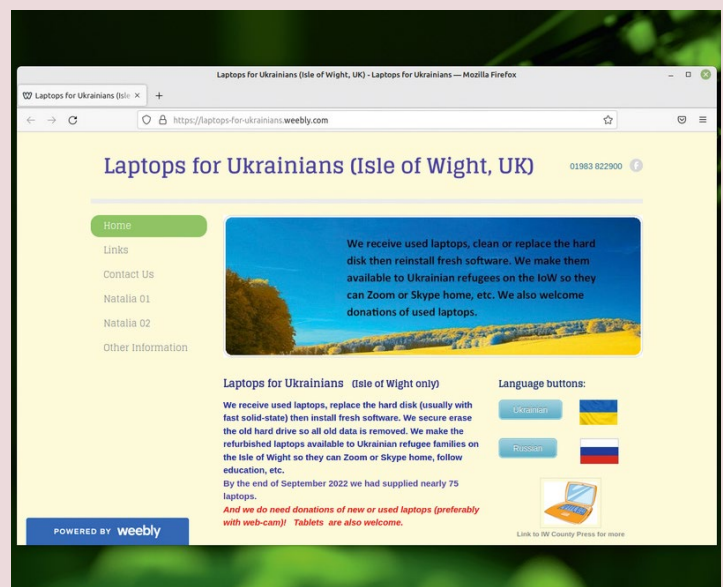
Most Ukrainian guests are mothers arriving with mobile phones, backpack luggage and children, staying with generous hosts across the Isle of Wight. A mobile phone is never sufficient for children's online education at home or necessary complex form-filling.

We only ever just get by with donated laptops, so more are always welcome. We'd also love to hear of any similar activity on the mainland. Especially in the Linux community, because Linux is a total gift for an activity such as this. Please get in touch via our website contact form (<https://laptops-for-ukrainians.weebly.com>).

Roger Skidmore, Isle of Wight

Neil says...

What a fantastic project to undertake and well done for helping so many people. If anyone is interested in donating equipment or creating something similar in their area, Roger is more than happy to help with advice, so do get in contact with him.



■ Linux can help refresh ageing hardware, so it can go on to have a useful second life.



shane_collinge@yahoo.com

SUBSCRIBE Save money today!

SUBSCRIBE TO LINUX FORMAT MAGAZINE

Subscribe today and enjoy great savings!

<http://www.magazinesdirect.com/LIN/B23R>

Call 0330 333 1113 and Quote B23R

**NOT FROM
THE UK?**

Turn to page 69
for more great
subscriber
deals!



Save money today! **SUBSCRIBE**

1) Only available to www.magazinesdirect.com subscribers

» **PLUS:** Exclusive access¹ to the *Linux Format* subs area!

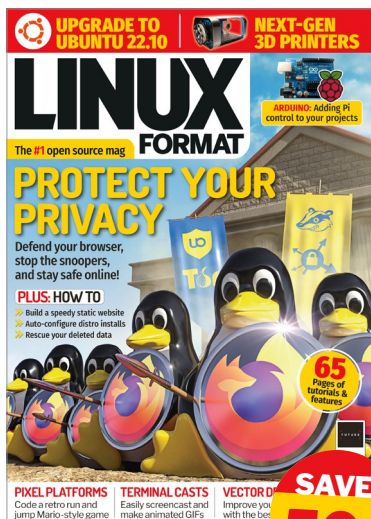
1,000s of DRM-free PDF back issues and articles! Get **instant access** back to issue 66 (May 2005) with tutorials, interviews, features and reviews. At **linuxformat.com**



» CHOOSE YOUR PACKAGE!

Our simple three-month best offer breaks up the annual subscription into four easily manageable payments over the course of the year.

ANNUAL PRINT EDITION



Only
£10.54
every 3 months

13 issues of *Linux Format* in print over 12 months and save 50%!

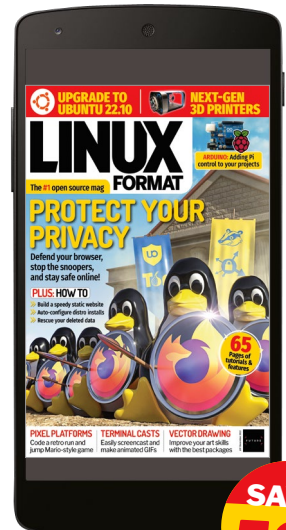
ANNUAL PRINT + DIGITAL EDITION



Only
£16.79
every 3 months

13 issues of *Linux Format* over 12 months in print and digital, and save 61%!

DIGITAL EDITION ONLY



Only
£11.35
every 3 months

13 issues of digital *Linux Format* over 12 months, and save 50%!

*Terms and conditions: Offer closes 28/2/2023. Offer open to new UK subscribers only. Pricing is guaranteed for the first 12 months and we will notify you in advance of any price changes. Please allow up to six weeks for delivery of your first subscription issue (up to eight weeks overseas). The full subscription rate is for 12 months and includes postage and packaging. If the magazine ordered changes frequency per annum, we will honour the number of issues paid for, not the term of the subscription. For full terms and conditions, visit www.magazinesdirect.com/terms. For enquiries please Telephone: +44(0)330 333 1113 (open Monday-Friday, 8:30am-7pm, and Saturday, 10am-3pm UK time, or email: help@magazinesdirect.com). Calls to 0330 numbers will be charged at no more than a national landline call, and may be included in your phone provider's call bundle.

JANUARY
Savings

Subscriptions from £/\$/€3*

SPORT



TECH



MUSIC



HOBBIES



SUBSCRIBE TODAY

SEE THE ENTIRE RANGE ONLINE AT

www.magazinesdirect.com/JS23

*TERMS AND CONDITIONS: Offer closes 19/02/2023. Offer open to new subscribers only. Savings are based on the newsstand cover price. After your trial issues, standard subscription pricing will apply. Please allow up to six weeks for delivery of your first subscription issue in the UK, and up to 8 weeks for overseas. Payment is non-refundable after the 14 day cancellation period unless exceptional circumstances apply. For enquiries, please call: +44 (0) 330 333 1113. Lines are open Monday-Friday 8:30am-7pm, Saturday 10am-3pm UK Time (excluding Bank Holidays) or email: help@magazinesdirect.com. Calls to 0330 numbers will be charged at no more than a national landline call, and may be included in your phone provider's call bundle.

REVIEWS

Wix

Shashank Sharma takes the feature-rich site builder for a test drive.

IN BRIEF

With hundreds of templates, support for over two dozen languages, and hundreds of widgets for integrating third-party sites and services on to your website, Wix is one of the most thorough website builders you're likely to find.

Wix is a giant in the website builder industry. Browse its site and clues to its success appear right away. While some services give you only a few templates, Wix has more than 500. Its built-in image editor has 40 Instagram-style filters. The blog supports 26 languages. The App Market has more than 200 widgets to enhance your site and integrate third-party sites and services.

Beginners can sample much of this for free, but experts can go much further, with features like Wix Code, enabling them to manage database collections, generate dynamic pages and custom forms, use their own JavaScript, access APIs and more.

It's not perfect, though. The templates aren't responsive in the usual sense, although the effect is very similar (Wix generates a separate mobile version of the site). Some features, like form building, come in the form of apps that require payment to get their full functionality. But nonetheless, it is a powerful and comprehensive service with a lot to offer all users.

Build a site

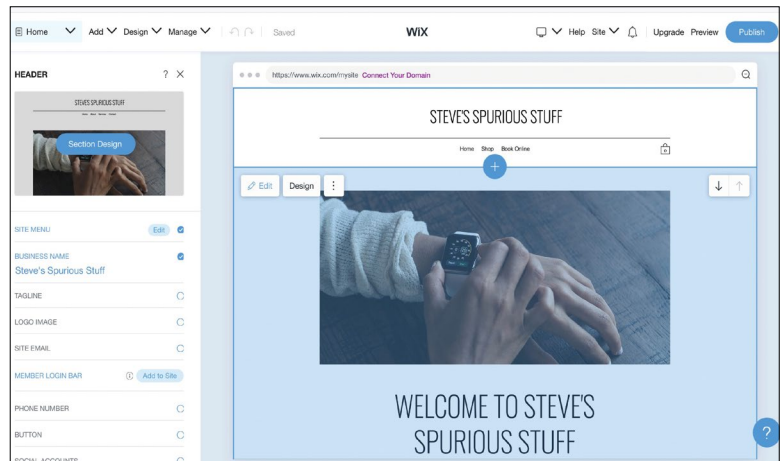
Account creation is as simple as entering your email address and a password. You're asked to choose the type of site you'd like to create – business, music, photography, blog and so on – and then whether you'd like to have Wix generate the site for you or if you'd rather do it yourself from scratch.

Opting for the regular editor gives you complete control of layout. Objects can be positioned with pixel-level placement, rather than being automatically aligned in blocks or columns. You can drag and drop complex blocks (contact forms with headings, buttons, text) as a whole, change alignments, even ungroup items within a block to rearrange them how you like.

The core editing functions are equally well designed and more like a native app than the average website builder. Right-click menus show relevant commands for different controls, for instance. Alignment guides and on-screen displays of control heights and widths help you position and size items accurately, and floating toolbars give speedy access to key options.

Ecommerce

Building a web store with Wix is simple. Browse an array of templates, add any extra components you need, and the visual previews and sample images make it easy to define the look and feel of your store. Don't be fooled, though – there's plenty of power under the hood. Wix supports digital as well as physical products,



A two-tier editor enables you to just get on with your site or do advanced pixel-level editing.

for instance, and also includes services. Products can be illustrated with videos, as well as images. You can give them custom options (size, colour), a weight or a SKU. Wix supports plenty of payment providers, too.

Plans and pricing

The free plan includes Wix branding, gives you access to the drag-and-drop editor, and in theory supports an unlimited number of pages. In practice, a 500MB storage limit and maximum 500MB data transfer per month mean it's only suitable for very small sites.

The Connect Domain plan, at £4 a month, lifts your bandwidth limit to 1GB but still includes Wix branding. Further plans increase the bandwidth and storage, and ditch the branding, right up to the VIP plan at £21 a month, with 20GB of storage and priority response support. There are three Business and eCommerce plans, too, which allow you to sell products online.

Wix's attractive templates and excellent editor make it easy for even total newbies to build a great-looking, powerful website. We're less convinced that it has the support services big sites require, but the core features are so strong that you need to try it. **LXF**

VERDICT

DEVELOPER: Wix

WEB: <http://wix.com>

PRICE: Multiple plans, starting with a free tier

FEATURES	9/10	EASE OF USE	8/10
PERFORMANCE	9/10	DOCUMENTATION	8/10

Despite its range of offerings, it's still just as easy to use and welcoming for novices as it's a delight for professionals.

» **Rating 8/10**

AMD Ryzen 5 7600X

A beastly “budget” gaming CPU that **Christian Guyton** has found has one big temporary drawback.

SPECS

Socket: AM5

Process: 5nm
TSMC

Cores: 6

Threads: 12

Clock: 4.7GHz
(5.3GHz boost)

Cache: 384KB
L1, 6MB L2,
32MB L3

Unlocked: Yes

GPU: AMD

Radeon two-
core RDNA 2,
400MHz

Mem max:
128GB DDR5-
5200, ECC
support, two-
channel

PCIe: v5

24 lanes

TDP: 105W

A MD isn't messing around this time; Intel's nifty Alder Lake core architecture put Team Blue in front in the current generation, and with 13th-gen Raptor Lake chips on the way, AMD clearly knew it had to swing for the fences. AMD is giving us a similar generational upgrade, but without resorting to 'efficiency' cores to save power; the result is a chip that dominates in single-core workloads, and still performs reasonably well when firing on all cores.

The new 5nm manufacturing process used by TSMC to produce the Ryzen 7000 chips is clearly excellent – and that's even bearing in mind that Ryzen 5000 was an extremely solid generation of CPUs. It's worth noting, though, that Zen 4 comes with a new socket, meaning AMD is finally leaving the trusty AM4 socket behind and mandating upgrades to new AM5 motherboards.

Socket to 'em

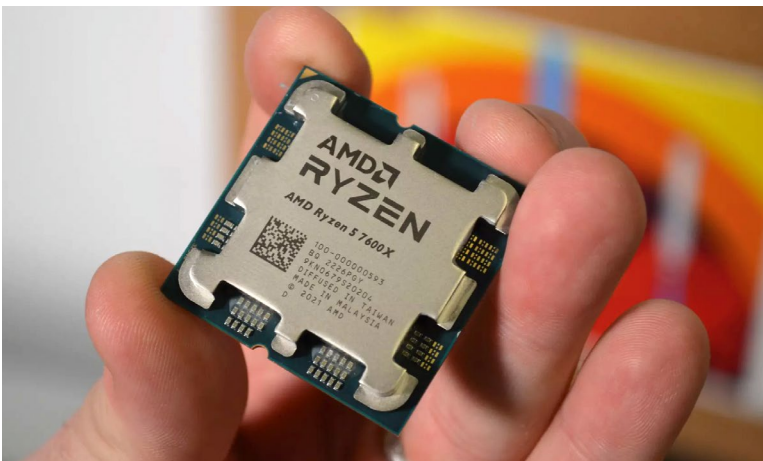
At £265, the Ryzen 5 7600X doesn't represent a generational price increase when compared to the ever-dependable Ryzen 5 5600X. The problem with the 7600X's excellent price-to-performance ratio is that it doesn't take into account the cost of upgrading the rest of your system to be compatible. These new processors require a Zen 4-supporting motherboard, with the new AM5 socket, add on top of this, AM5 boards only support DDR5 memory, so you'll be out of pocket for some new if faster RAM, too.

This isn't entirely AMD's fault; Team Red had to move on from the AM4 platform eventually, but with even the cheapest X670 AM5 motherboard costing £270 and DDR5 memory costing a little more on top, it's all extra costs you have to factor in.

With six cores, 12 threads, and a boost frequency of 5.3GHz, the Ryzen 5 7600X is no slouch, with a step up of 0.6GHz at boost from the Ryzen 5 5600X, and the upgrade also comes with much faster PCIe 5.0.

Naturally, those new compatibilities mean better performance – and while Intel users have had access to them in 12th-gen Alder Lake chips since late 2021, the market hasn't really reflected that; PCIe 5.0 SSDs might actually be available to buy at some point in early 2023, and Intel's decision to keep Alder Lake backwards-compatible with DDR4 has evidently led many buyers to stick with their existing memory.

The 7600X comes with integrated graphics, running a dual-core RDNA 2 GPU – this isn't going to scare any dedicated graphics card, but it's a worthy inclusion



Finally AMD has the goods to take the processing crown back from Intel, at least for a while...

considering the applications in GPU-less systems like lightweight laptops and office PCs. There's AV1 and HEVC decode, and support for DisplayPort 2.0, HDMI 2.1 and USB Type-C video output, which all keep this chip competitive with Intel's forthcoming offerings. Another inclusion clearly designed to keep pace with Intel is support for Intel's AVX-512 instruction set.

One thing the 7600X doesn't have is a dedicated cooling solution, possibly marking the death of AMD's excellent Wraith line of stock air coolers. The upside is that AM4-compatible coolers work on AM5 boards, so you might not need a new cooling solution.

General processor single-core performance pegs the Ryzen 5 7600X a few percentage points ahead of the formidable Intel Core i5 12600K and a good 22% faster than the Ryzen 5 5600X. This plays out to on average the Ryzen 5 7600X outpacing its older model by 35%, matching the Ryzen 9 5950X and outpacing the Core i5 12600K. With a peak max of 111W, it consumes more peak power than the 5600X's 78W, but on average used 10W less across tests. **LXF**

VERDICT

DEVELOPER: AMD

WEB: www.amd.com/ryzen

PRICE: £265

FEATURES	9/10	EASE OF USE	9/10
PERFORMANCE	9/10	VALUE	8/10

AMD is coming out swinging with its new generation of Zen 4 Ryzen processors, with the 7600X providing an exceptional level of performance for a more affordable CPU.

» **Rating 9/10**

WattOS R12

Nate Drake explores how WattOS can bring your ancient hardware to life with this zippy, if spartan Linux distro.

IN BRIEF

After six years in development limbo WattOS R12 is finally here. Now based on Debian 11, it still focuses on ease of setup and speed, running comfortably on older machines.

SPECS

CPU: 1GHz
64-bit
Mem: 512MB
HDD: 10GB
Builds: 86_64

We love lightweight distros. This is why we reviewed some of the best of them in **LXF295**, but WattOS didn't make the cut, because a new version hadn't been released since 2016.

This is most likely because WattOS is the creation of just one man: Ronald Ropp, aka Biff Baxter, who started development of the distro in July 2008. Unlike previous versions, WattOS R12 is based on Debian, not Ubuntu, which in theory makes for a much more stable operating system. This is reflected in the bundled apps, such as *Firefox 102.4 ESR* (Extended Support Release). Using Debian alone usually means you have to do extra tinkering to install the latest versions of programs, but WattOS has taken care of this for you, supporting installation via both Flatpak and Debian backports.

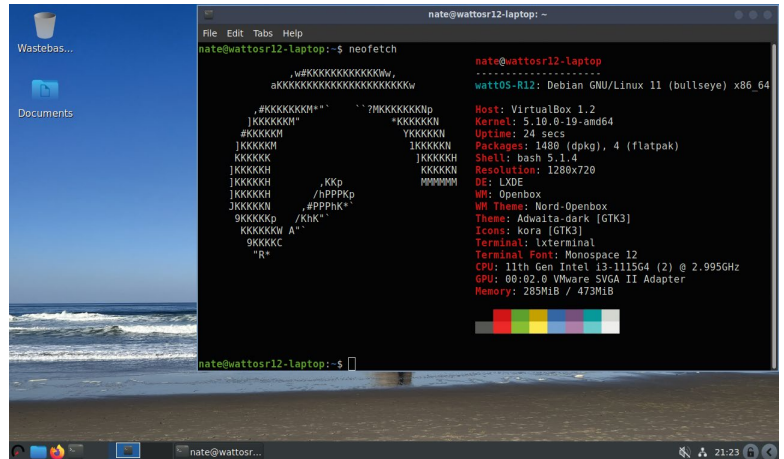
Set up is very simple via the *Calamares* installer and takes only a few minutes. This is one of the fastest OS installations we've ever done. The lightweight LXDE 11 desktop environment is extremely quick to respond and the install footprint is just 5GB, putting even minimalist distros such as *Lubuntu* to shame.

According to contributors on WattOS's Discord server, the OS can run on some incredibly ageing machines, as well as budget Chromebooks. When testing in a virtual machine, we found it ran perfectly well on Debian's minimal stated specs of 512MB RAM and a 1GHz CPU. There's no specific ISO for running WattOS on old 32-bit machines, however.

This minimalist ethos carries over to the pre-installed apps, which are very spartan. For instance, the only bundled office software is the *Evince* document viewer. Internet apps comprise only the aforementioned *Firefox ESR* and version 3.0 of the *Transmission* BitTorrent client. You can also play media files via *VLC 3.0.17.4 (Vetinari)*. Beyond this, it's largely down to you to install any other programs you need.

Installing *LibreOffice* via Flatpak wasn't too troublesome but we had to open *LXTerminal* to set up the relevant software plugin so that installation via the FlatHub website could be done graphically. This was a minor niggle, as it only had to be done once, but suggests that WattOS may not be for users who want to avoid the command line altogether. In fairness, we could just as easily have installed *LibreOffice* and other software using the GUIs of the pre-installed *Software Center*, *GDebi* or *Synaptic* package manager.

Although the R12 release notes don't explicitly make this assertion, WattOS has traditionally claimed



A custom version of the minimalist LXDE desktop environment makes for a snappy interface.

to be energy efficient. Given the very small number of running processes, it didn't surprise us that our version of R12 only drew down just under 1.4W in a virtual machine. Still, before wiping your existing install, consider testing WattOS in live mode to see whether it really will extend your laptop battery for hours.

There's no dedicated wiki or manual, but if you run into difficulties using WattOS, help is available via a well-attended Discord server. The forum community is also extremely willing to address users' issues. This might be enough to persuade less-experienced Linux users to take the plunge with this distribution, given that they may have to attend to some post-install tweaks themselves.

Overall, if you like your distros super-fast and barebones, and are happy to manually install only those programs you need, then WattOS R12 is the right distribution for you. Because WattOS is based on Debian 11, which is supported for five years, there's also little danger of it becoming obsolete, even if the good people at Planet Watt drag their heels about releasing a new version in the future. **LXF**

VERDICT

DEVELOPER: Ronald Ropp
WEB: www.planetwatt.com
LICENCE: GNU GPL and others

FEATURES	6/10	EASE OF USE	9/10
PERFORMANCE	10/10	DOCUMENTATION	7/10

WattOS's no-frills interface is clean and responsive, but it may need some work before you have all the apps you need.

» Rating 8/10

Kali Linux 2022.4

If you exist in the grey twilight between good and evil, and want to look cool while doing it, **David Rutland** has just the distro for you.

IN BRIEF

Kali Linux is a mature and stylish distro, which has all the tools you need to pursue a career in cybersecurity, ethical hacking or penetration testing. New and updated tools add to already impressive capabilities, and it runs on practically everything – including the Raspberry Pi and specific smart phone builds.

SPECS

CPU: 2GHz
Mem: 512MB
HDD: 2GB
Builds: i386, x86_64, armhf, armel, arm64

Kali Linux is an evil distro for hackers, ne'er-do-wells and other undesirables who want to overthrow the state and sow chaos. You can tell because of the menacing wallpapers that feature evil-looking dragons, who are doubtless up to no good.

There's also the vast array of tools designed for network analysis, penetration, password cracking and other such things, which we'll get to shortly. They're kind of a giveaway.

Rapidly approaching its tenth birthday, Kali is the go-to distro for individuals in the network security field, those wanting to get into the field, and people who want to look cool and mysterious while hammering on a keyboard in the local Costa Coffee.

The 22.04 release just squeezed in before the end of 2022, and in recognition of the essential nature of the ubiquitous Raspberry Pi, Kali has now been added to the *Raspberry Pi Imager* tool, so it's simple to install this hacker's toolkit on proper DIY hacker hardware.

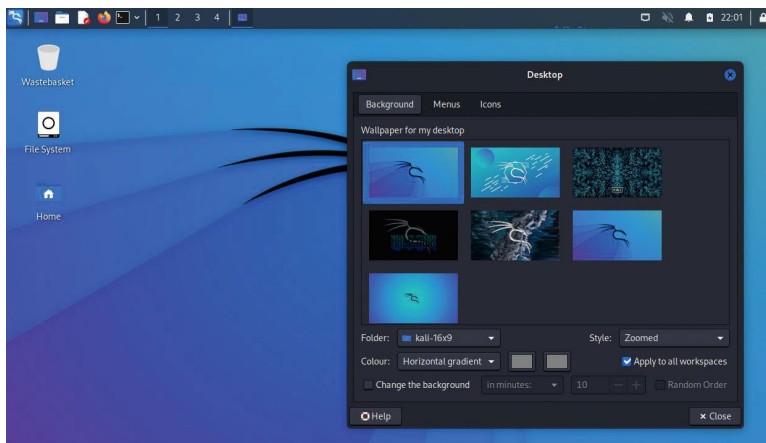
Unfortunately, Pis are in short supply, so we had to make do with installing Kali the old-fashioned and proper way: on a virtual machine. You can also install Kali on WSL, as a live USB, in a docker container, as a vagrant image, as a ready-to-go app on various cloud providers, or even on certain mobile phones. Hardware resource access, and other more esoteric factors, may vary, but Kali makes it easy to install on anything – with dedicated images for each, so you can maximise the resources of what's available.

Kali is based on Debian, so installation on our decidedly non-exotic VM was simple and familiar, once the epic download of up to 11GB was out of the way. There were no nasty surprises or unnecessary complications, and after selecting Xfce over either Gnome or KDE, we were on the desktop in a couple of minutes, staring into the side-eye of a malevolent dragon who wanted us to hack something – anything!

Because, don't get us wrong, unauthorised entry is Kali's raison d'être. Sure, you can use it as a daily driver, and it's great. You get everything you'd expect with a minimum Debian install – plus a whole lot more. You can install *LibreOffice*, *GIMP* and *Steam* without too much trouble. But that's not what Kali is about.

New tools for mischief

The central tool suite of Kali has been more or less the same since its inception, but there are refinements in this latest release. Particularly notable are updates to the wireless firmware, and better wardriving



There are nine evil dragons of hacking in this screenshot. We bet you can't spot them all. Download the ISO and find them yourself.

Bluetooth, SDR and MouseJack support – just what you need when penetration testing.

There are some additions, too. These include **bloodhound.py**, a Python-based ingestor for *Bloodhound* (a tool for finding hidden relationships within an active directory or Azure environment); **peass-ng**, a collection of privilege escalation tools for Windows, Linux and Mac OS; and *Idapdomaindump*, which does exactly what you probably think it does.

More shiny for your eyeballs

Everyone knows that 1337 haXors are all about the cool eye-candy, and Kali doesn't disappoint. Desktop-wise, Gnome lovers will be pleased to find Gnome 43 as an option. You can also deploy your exploits from KDE Plasma 5.26. As always, the wallpapers are top-notch artwork, usually featuring dragons. There's also a graphical network monitor in the task bar – neat.

All you need is a trenchcoat and some RGB, and you can be the hero of your own cyber action movie. **LXF**

VERDICT

DEVELOPER: Offensive Security

WEB: www.kali.org

LICENCE: GPL3

FEATURES 10/10
PERFORMANCE 8/10

EASE OF USE 9/10
DOCUMENTATION 9/10

Even without its killer suite of ethical hacking tools, Kali is the ice-cold epitome of style and substance, which is as easy to use as you'd expect from a Debian derivative.

» **Rating 9/10**

Nitrux 2.5.1

An impressive beginner distro that's well worth your time, even if you spend most of it tweaking the settings, says **Mayank Sharma**.

IN BRIEF

Targeting beginner Linux users, Nitrux does a splendid job of simplifying software discovery and offering a slick KDE-based interface, but perhaps leans too much into the minimal design aspect.

SPECS

CPU: 2.5GHz
dual-core 64-bit
Mem: 2GB
HDD: 5GB
GPU: 32MB
VRAM,
OpenGL 2.1
Builds: x86_64

If you are just taking your first steps in Linux, you should begin your journey with Nitrux. It leverages some of the most stable and well-respected open source projects and builds on them to create a distro that suits the sensibilities of new Linux users.

For instance, instead of building a custom desktop environment from scratch, Nitrux relies on KDE Plasma's malleability along with some redesigned components to simplify workflow for new users. It offers a couple of layouts for the desktop to enable users to make best use of the screen real estate.

Nitrux has even customised some admin apps to make tasks such as firewall and backup approachable for inexperienced users. It ships with a good dose of apps for all kinds of desktop tasks.

The distro also supports and encourages the use of Applimages, makes it straightforward to install extra apps. In fact, Nitrux plans to eventually switch all its graphical apps to Applimages. These are launched in a Firejail sandbox, which makes Nitrux more secure out of the box than your typical desktop Linux.

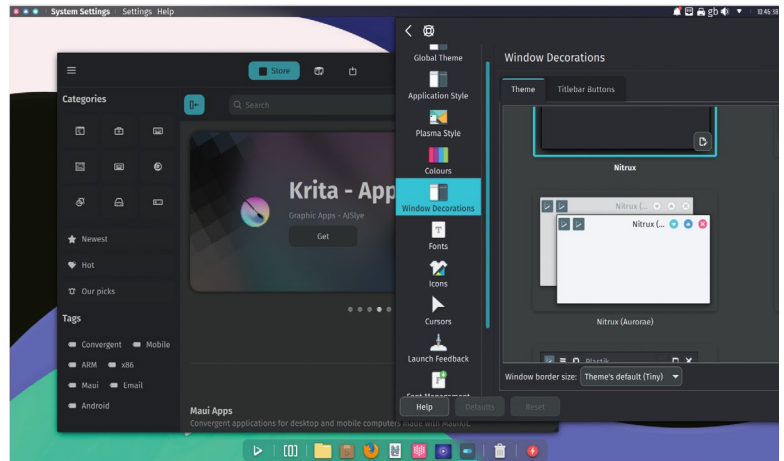
Easy does it

Trying Nitrux is a snap. The 2.8GB downloadable ISO served from a speedy dedicated server offers Live booting, so you can try it out easily and, once happy, install it from here, too. The *Calamares*-based installer is clear and inviting, and makes setting up keyboard and passwords, plus partitioning with encryption, obvious and clear, though it seems to default to manual partitioning, which isn't ideal for beginners.

Once into the custom KDE desktop, the main sticking point is the unified window controls that permanently live top-left of the desktop, rather than as part of the active window. So, if you want to minimise, expand or close the current window, you need to override your muscle memory and shift the mouse to the top-left. It's somewhat baffling and you can't even double-click the title bar to expand or contract.

KDE is renowned for its customisability and you can alter many of these default behaviours and styles through the Windows Management and Appearance settings, which also offer comprehensive window tiling features. It's just a matter of discovering these and tailoring them to your needs, but it feels like an option that could be put into the initial *Calamares* installer.

The design oddities continue through to the software Store – you have to decipher the title-less category icons or hover the mouse over them to get



Sitting somewhere between Ubuntu's minimal Gnome desktop and Mint's traditionalist design.

your pointers. The interface uses disappearing scroll bars, so you won't know they're there and you can't just jump back to the root of the Store. That aside, the Store works a treat and the use of Applimages means managing software is entirely straightforward.

The distro has an active community that you can interact with on all the popular social networks. However, the lack of a dedicated documentation section is a downer, there is a tutorial section that offers guides to many general topics, such as *Samba* shares, using *Wine*, *Wayland* and more.

We're encouraged by the regularity of updates and speed of response by developers to queries. This latest 2.5.1 update, released in early December, brings support for the latest Mesa 23.0 release and introduces Kernel 6.0. Nitrux is available for 64-bit computers only, but can also boot on older hardware with the Legacy BIOS, as well as newer EFI machines.

Certainly worth your attention just for the speed of updates and number of customisation options, which are the source of our main confusion, and could be reduced with an installer-based option. **LXF**

VERDICT

DEVELOPER: Nitrux
WEB: <https://nxos.org>
LICENCE: GPL, Various

FEATURES	7/10	EASE OF USE	8/10
PERFORMANCE	8/10	DOCUMENTATION	6/10

It's slick, fast, regularly updated and offers just enough documentation but the KDE desktop roots can confuse.

» **Rating 8/10**

Victoria 3

“The peasants are revolting,” cries The Management, but really it’s just **Jonathan Bolding** serving up their afternoon tea and biscuits.

SPECS

Minimum

OS: Ubuntu

20.04 LTS

CPU: Intel Core

i3 3250, AMD

FX 8370 – AVX

required

Mem: 8GB

GPU: GeForce

GTX 660,

AMD R7 370,

2GB VRAM

HDD: 10GB

APU: Intel HD

Graphics 630,

AMD Radeon

Vega 8

Recommended

CPU: Intel Core

i5 6600K, AMD

Ryzen 5 2600X

Mem: 16GB

GPU: GeForce

GTX 1660,

AMD RX 590,

6GB VRAM

It’s 1836 and the Industrial Revolution’s machines are in high gear. The coming century will change the face of the human world with a population explosion, a second industrial revolution and more. In *Victoria 3*, you take control of a society attempting to ride this wave of explosive change to its end in 1936. What you do along the way, and how much fun you have, is up to you.

Victoria 3 tasks you with building a nation by shaping its laws, economy, people and institutions. Developer Paradox calls it a “society-builder”. It wants you to care about the minutiae of political movements, distribution of power, population trends, economic organisation, factory output or global trade, and then start manipulating them.

That might be hard. *Victoria 3* boasts some of the most intricate game mechanics we’ve seen in a strategy game. They can be overwhelming. You can micromanage trade routes, chart paths for societal reform, tweak your nation’s build queue, or examine production methods. Even a small nation is constantly given options to fiddle with, and progress in large nations can slow to a crawl as you scramble between trade crises and multi-front wars.

Managing this is made easier by one of Paradox’s better interfaces. It’s not quite as friendly as *Crusader Kings 3*, but it does have significantly more complex systems to work through. The bottom bar in particular is useful, with shortcuts to all kinds of national actions, from building and diplomacy to mustering troops. The map, meanwhile, is striking and impressively detailed.

I am the law

It’s in *Victoria 3*’s simulation of how a society is governed that the game is at its best, drawing you deeper and



deeper in. Governments are based on laws: sets of options that cover diverse topics such as how your economy is structured and who gets to vote. Changing laws requires the support of interest groups. *Victoria 3*’s finest mechanic. These groups, such as landholders, petite bourgeoisie, armed forces, trade unions, industrialists, religious authorities and rural folk, gather up clout from their members and try to push their agendas, forming dynamic political parties.

These groups intersect masterfully with every other game system, creating opportunities and obstacles. Maybe you want to change tax laws to fix your budget, but the powerful industrialists will cause a big stink, or maybe even team up with another group and threaten revolution. So, you change voting laws to give the poor power, reducing the industrialists’ political clout. Of course, this means you now have another group able to push forward their agenda.

The core of all those interest groups are people, also known as pops. Every person on the planet is simulated in the background. They have jobs and families, and they want various necessities and luxuries (like heating? – Ed). They migrate and move in a system that is fascinating to watch. When ideologies such as socialism and fascism are invented, pop groups gain supporters or goals, triggering unrest and revolution at home and abroad.

Your big concern with pops is their standard of living, cultural discrimination and employment, but trying to improve these involves solving big social and political puzzles. Freedom of religion and giving immigrants citizenship annoys other people even while it improves lives, so it’s a thorny but compelling balancing act.

Big business

Alongside managing a nation’s people, your hand is also on the tiller of trade and industry. *Victoria 3* has you serve as a kind of omniscient chief executive, deciding what new factories, mines, infrastructure and farms get built. You need cash for that, of course, but whether it’s

Who doesn’t like a good map?



from wealthy investors or the state itself, these are your decisions – the buck always stops with you.

Victoria 3's economic system is a slick simulation that incentivises real-world behaviours. You don't always want to hoard wealth. You can run a deficit for a few years, then build up a reserve, or nearly bankrupt yourself fighting world wars before entering years of austere recovery. And if you're missing some important goods, you can look abroad and exploit less developed nations.

Planning your economy, finding out what goods are needed to industrialise, what raw inputs to source and from where is the stuff that economic strategy dreams are made of. The added complexity of deciding production methods and ownership stakes is just icing on the customisable cake of your nation's market.

As opposed to fiddly tracking of every single bushel of wheat, *Victoria 3* provides comprehensibility by saying that while supply of materials is unrealistically infinite, it'll eventually hit a very, very expensive price ceiling or become so cheap its market crashes into the floor. Helpfully, a rich trade system lets you quickly address deficits or excess from the same screen, and feedback notifies you when trade routes post a negative balance.

The intermingling of business and state quickly becomes a colonialist machine that eats people and spits out bones. You'll fight wars of conquest over rubber just so your people can have bicycle tyres – or find your budding socialist utopia playing world police to stop others from doing the same.

Aggressive negotiations

Eventually, of course, you have to deal with conflicts both domestic and international. These aren't always armed, however, and many take place across the negotiating table. This is perhaps the weakest part of *Victoria 3*, and only a few key systems keep it from being a total dud.

Militaries are managed abstractly, with your armies run by generals assigned to fronts. You can tell them to hold the line or try to advance and capture territory. It's a pretty fitting simulation of 19th century warfare – at least from a politician's chair – because it's about hoping the army you've built, the equipment you've produced, and the person you've picked to lead it can bring you victory.

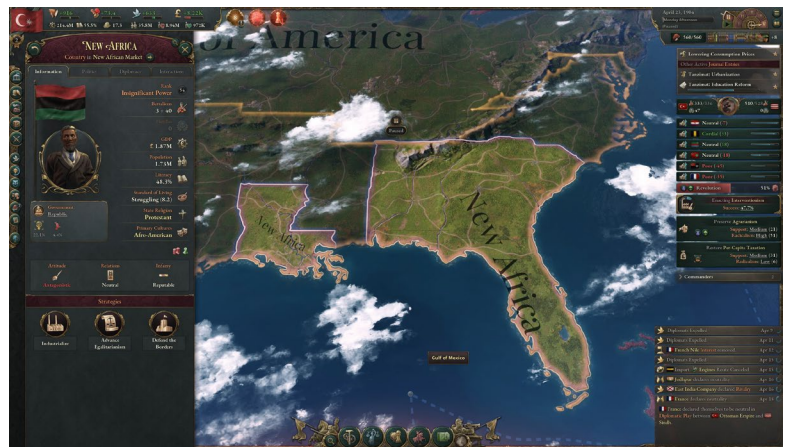
Your suite of diplomatic options are simple. In order to do most things, you make a diplomatic play: claim a state controlled by another country as yours, demand another nation submits as a puppet regime, force others to adopt specific laws. You make demands, the other side makes counter-demands, and outside nations either stay neutral or get bribed to pick a side. You raise and deploy

» ARTIFICIAL UNINTELLIGENCE

Smaller nations without a sizable army can get locked into civil wars for an interminably long time, rendering chunks of the map unusable unless you want to declare war and end it yourself. Bigger ones have the same rich inner life as yours, but lack the guiding hand of a person aiming them at bigger goals.

That's exacerbated by the AI's short memory. In a game playing as the Socialist United Syndicates of America, we helped every communist revolution across the globe to power. More often than not, within a month of winning victory, those newly forged sibling nations would decide we were an existential threat and adopt a massive penalty to relations with us.

That's the exception, though. If anything, the nations of *Victoria 3* are too reluctant to forge diplomatic pacts or start wars. You'll never see anything resembling major conflicts such as the Franco-Prussian or Crimean Wars, let alone an international conflict like World War I.



troops to rattle your sabre. If nobody folds before the timer runs out, that means war.

It's a tidy system, but the diplomatic actions around it are anaemic and difficult to pull off, because when it comes to international relations, the AI is static or even unpredictable. Sometimes it stays friends for decades, or it turns into hateful a rival overnight. It's not clear if this is down to bugs or invisible changes in other countries.

Victoria 3 is weak on the international stage, but if your fantasy is to take control of a nation and set goals, build a society and play around with complex systems, it fulfils that brilliantly. The internal life of your nation is rich, and for much of the game that's where are – up to your neck in domestic politics and squabbling pops. **LXF**

It all feels familiar and yet somehow different.



VERDICT

DEVELOPER: Paradox Development Studio

WEB: www.paradoxinteractive.com

PRICE: £41.99

GAMEPLAY	8/10	LONGEVITY	8/10
GRAPHICS	7/10	VALUE	6/10

Attractive historical sandbox filled with potential, but it's on you to unlock it if you can work around iffy war mechanics.

» Rating 7/10

Roundup

Element 1.11.14 » Cinny 2.2.2 » Mirage 0.7.2
» NeoChat 223.09 » Fractal 4.4.0



Mats Täge Axelsson

keeps learning more about Linux, even after all these years. How many? Guess!

Matrix chat clients

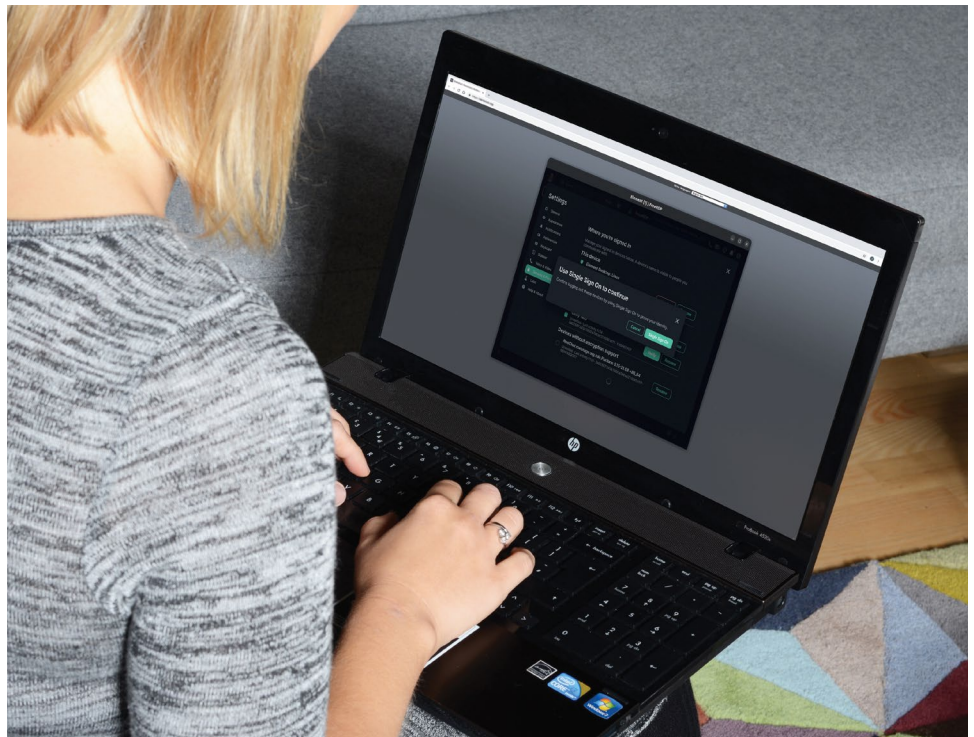
Mats Täge Axelsson examines the potential of distributed messaging and federated authentication servers that use the Matrix protocol.

HOW WE TESTED...

We paid a visit to <https://matrix.to> and got ourselves an account. If you want to try it for yourself, you can get an account there or somewhere such as <https://riot.im>. Ideally you should use a password-supported account as many clients do that support SSO or OAuth2. Next, we went to <https://matrix.org> to find clients.

Installing them on Linux was mostly a breeze thanks to the projects supporting plenty of distros and packaging standards. Figuring out how to log in and when you can use something other than password authentication was the first hurdle. After you've done it a few times it becomes obvious, but it differs in some details between apps and sites.

Lastly, we realised that in order to keep the account secure, there are several layers of functionality that you can deploy if you want to be diligent – which you should be, as one point of this network is privacy.



Having seen how Big Tech tracks everything you do in the name of profit, you may be tempted to use an alternative means of communication. Platforms such as Facebook and Twitter don't seem to be improving from a moral point of view and are becoming more and more crowded with ads. To counter this, alternative platforms have appeared.

Most of these solutions fall short if they are run by another big business that does everything in the name of money. Instead, you can choose to put in a little more effort and have your own username and maybe

even server. For this to have a chance to work, standards are a necessity. One such standard is Matrix, developed and maintained by the Matrix Foundation, a group you can support yourself if you wish.

Following the protocol, you can use any of the mainly free and open source chat apps available, or write your own. To get started, you need to register an account on a server, the most popular being <http://matrix.org>. You can also set up your own server, but we're only talking about clients here. You can use OAuth2 authentication to get an account but not all apps support this yet.

Ease of installation

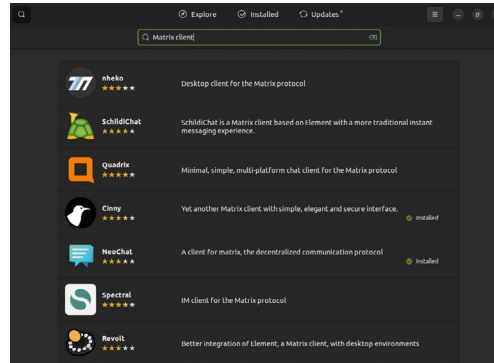
Most applications offer multiple package formats for installation.

Being chat applications, Matrix clients often come as web services. An example of this is *Cinny*'s web page, which has a link to the online version of the application. You have to actively seek out the Flatpak edition. This is the same for *Element*, although installing it on your smartphone is actively encouraged. It doesn't mean it's impossible to install these on Linux, it's merely a quirk of their web pages.

When you start looking around, you will find that downloading the desktop application is second choice for several of these clients. But that makes it more interesting to have them on your desktop – you are a rebel, right? The ones that don't do this are *Fractal* and *NeoChat*; the teams behind these are part of the KDE and Gnome teams, making them lean in their obvious preferred desktop directions. Finding packages for *Fractal* is easy because it exists on Flathub. *NeoChat* comes as a Snap and a Flatpak. The nightly version also comes as a Flatpak.

When you come to choosing packages, you'll find Flatpak is a common option. The best clients come as both AppImage and Flatpak, like *Cinny*. Another that offers both is *Mirage*. It goes a few steps further and also offers packages for Arch, Gentoo, Nix and others. On the GitHub page, there are also clear instructions on how to install it manually from source on more distributions.

Source install instructions are excellent for *Mirage*. In fact, the instructions clearly show how to get started developing a new



Searching in the Ubuntu software store for Matrix clients results in a long list of packages. Which one will you choose?

project. You must take care with *Element* because development happens in different Git repositories for different platforms. The desktop version can be found at <https://github.com/vector-im/element-desktop>. The project's main focus is on the web app, and all other versions stem from that.

If you go for a tar install, make sure you have the **libolm** library. This is one of the basic libraries required for the Matrix protocol. It covers the encryption part, which is essential for privacy. Oddly enough, some of the applications do not support encryption, even though they support private rooms. Possibly better known is the *Signal* application – the team behind this pioneered the algorithm and is still maintaining the development of the libraries.

VERDICT

ELEMENT	8/10	NEOCHAT	7/10
CINNY	5/10	FRACTAL	7/10
MIRAGE	9/10		

Packaging is becoming easier on Linux, despite the choices. In this respect, **Mirage** has the most options.

Look and feel

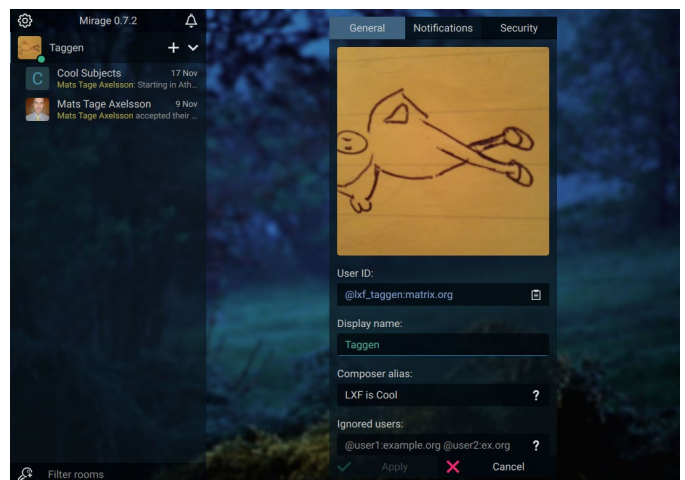
You'll see the interface a lot...

To withstand the pressure of the next flame war, your interface had better look good. But good looks are subjective. Some people want a lot of colour and a picture-based background, but a more basic issue is whether your text is dark or light. Most of these clients use the dark or light theme, and most have a total of four variants. *NeoChat* has four colour themes and some other tidbits including Bubbles and Compact for the message feed. In *Fractal*, you can change the theme to match the system's, other settings seem to be missing.

To make your view the best possible, you can change a lot more things in *Mirage*. Coming from a Qt toolkit, the application's appearance has been prioritised for the design of this package. *Cinny* is quite lacking when it comes to tweaking its looks, although you do have four colour options.

In *Element*, you have a choice of a light or dark theme, three message layouts and font size settings. Additionally, you can choose whether image messages are small or large.

All clients can notify your desktop in Gnome and most other standard desktops. They all allow On, Off or Noisy settings. To really make your experience visually stunning, you should use *Mirage*, it includes all those features and you can have a picture background. If you set it using theme files, you can really go wild.



The *Mirage* application has the flashiest interface of them all. Hopefully, development will take off again to add more features.

VERDICT

ELEMENT	8/10	NEOCHAT	6/10
CINNY	7/10	FRACTAL	5/10
MIRAGE	10/10		

Offering more than just dark or light themes, **Mirage** has the best interface aesthetics of the bunch.

Logging in and getting started

Having conversations on Matrix is open until you set up your own groups.

In many ways, you can use Matrix as a social media alternative. You have options to join groups and chat about your favourite topics, and talk directly with friends. One big difference is that on the Big Tech services, you are set to public by default and you can choose subjects as you go. With Matrix, you choose your subject by joining a group.

Another big difference is that if you want to have private conversations, you must take action yourself. You need to create a group where you alone can invite people. For it to be truly secure, you must also create encryption keys.

The strength of this is that you are responsible for your own security. You need to know how to verify any people who join your group. This requires tools that can help you handle security – and each client has one such tool.

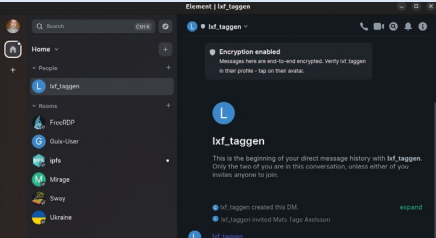
Element

9/10

When logging in using *Element*, you are greeted with a clear welcome screen and the option to log on with your username and password, or with SSO. On the password section, you have the choice of one standard identity server. Since there are already hundreds of servers, that is a little weak, although you can also enter your choice of server. SSO offers the usual suspects: Google, Facebook, Apple and the two Git options.

Element has a simple and clear interface where you can easily add your own groups and rooms. You can change settings for the size of images in chat feeds. Themes are limited to dark or light, and you can set the font size easily. Otherwise, theming has only a few standard options and no backgrounds.

Encryption for verification of other sessions and room encryption keys is simple if not convenient.



Cinny

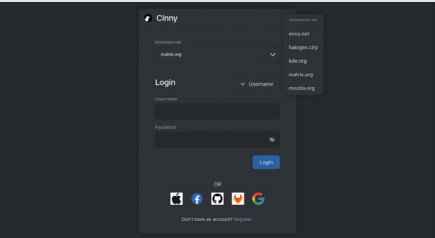
8/10

The login view of *Cinny* looks remarkably similar to that of *Element*, with one important difference: it has several homeservers in the list. You also have a theme even at the login screen. The phrase “heating up” is a nice quirky touch.

Groups and rooms look almost the same as in *Element* – again, there are just small tweaks. The People tab is on by default, which makes it easier to send direct messages. In the Settings window, you get a tab-organised screen, making it simple to find your various settings.

Encryption and other features work the same way as in the other clients here.

What *Cinny* has going for itself is that it is easier to make it look good and it's more comfortable to use than the others. Usage is so similar to *Element* that it comes down to personal taste as to which is best. Being second in this section only means that it's not our favourite.



Community and status

Use the power of community for yourself.

You want to communicate with someone about a particular subject using Matrix. Whatever topic you are looking for is probably available elsewhere, but with Matrix you can have direct conversations. For these packages, you can do what you always do: scour question sites where the answers have been waiting for a long time. This works, but some things are difficult to find; if you enter a chat interface, you can get hints where to go next.

Before choosing a specific client, you should go and see what is happening with the project. It is useful to know whether there has been any work done on the project over the past year before getting used to it. There are many ways to check this – one is to look through the last question on one of the Stack Exchange sites. The other is to check the developer's code site, which is usually GitHub.

Looking at that, you may notice that *Mirage* has had no new updates for many months but the room is very active. It means that the client is popular to use but development has slowed or

stopped. Hopefully, someone will take over the project, because it looks really nice and only needs the few features that are missing for full usage.

NeoChat, on the other hand, has daily code updates, so is still actively maintained, which means you can rely on it to keep working. *Fractal* is a quirky one – it seems way behind the competition but it has a very clear new goal with a lot of developers. *Cinny* has major updates coming and a large following, while *Element* is in very deep development mode, promising a brand new version.

VERDICT			
ELEMENT	9/10	NEOCHAT	8/10
CINNY	7/10	FRACTAL	7/10
MIRAGE	5/10		
Most of the projects we're looking at here are still in active development and maintenance, but Element is possibly the best supported.			

Mirage**7/10**

When logging in using *Mirage*, you have a screen that should show you the stability of servers and other pertinent data. However, this isn't working. You can only use a username and password combo – single sign-on from GitHub and the like is not yet supported. In the current version (0.7.2), the Register tab is there but it is not yet implemented. As for homeservers, there seems to be a plan for them but by default all you have is **Matrix.org** for now.

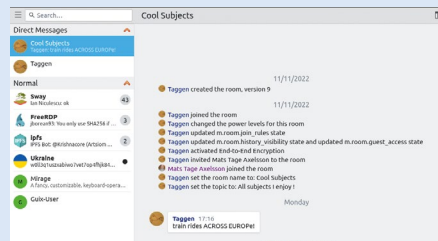
When it comes to organisation, you have the same groups and rooms like all the others, but they're implemented better. Rooms are tightly listed and you have a filter option. Inside the rooms, you can set the messages to have boxes. Next to the room, you can list all the participants or check the values of the room. Verification is lacking, though, which limits *Mirage*'s scope.

**NeoChat****6/10**

NeoChat's login has a single box where you put your username and homeserver before you can move on to the password. It's an odd way of doing this, without help for other homeservers, but it does stop you from sending your password until the server is ready to take it. You can use multiple accounts; they show up as tabs under the Rooms And Groups column.

Rooms are shown in a simple list, all according to the KDE standard. *NeoChat* has a list of participants in the far-right column, which also contains pinned messages from the room. You can change the look between Bubbles and Compact, and four themes of light or dark.

All the verification and encryption functions work, and settings are easy to find with the column view for features. Overall, *NeoChat* has the necessary features but lacks the fancy graphics that would make it more pleasant to use.

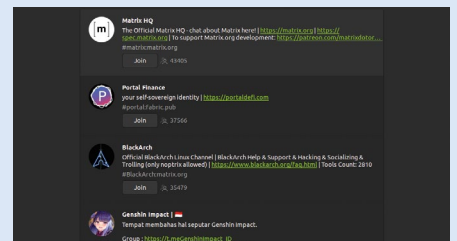
**Fractal****5/10**

Fractal does not offer any hints about homeservers – you have to supply your own. The login screen is plain but it works. There is no offer of using single sign-on, so you must get a password account. You need to go through two steps to get there, so it feels a bit awkward.

The opening screen is organised the same way as the other clients here but with a sparse interface. For colour, you can use the Gnome theme – *Fractal* is a native Gnome application.

As for organisation of the screen, that is also simple, without any fancy tricks or tweaks to make. It simply does what it is supposed to do.

This application also lacks encryption and verification of the session. Overall, the impression is poor and it needs to be improved, which the Gnome maintainers are working on in *Fractal 5* – the next generation is coming soon.



How are they made?

Language and framework make a long-term difference.

With the correct language and framework under the hood, new features can be added when the other projects add support. Having a code base that clearly has high activity ensures an interesting future for a project. It also shows you on which devices you can use it. You may prefer an application that looks the same on all devices, but with the wrong code base, this is unlikely to happen.

In this regard, *Element* sticks out because there are several clients for many platforms. Written mostly in TypeScript and JavaScript, *Element* easily converts to different platforms. The web page offers clients for Android, iOS and web. Obviously, there is also a Linux desktop client available.

The *Mirage* client, however, comes only as an AppImage and a Flatpak, which are excellent for Linux purposes, of course. Being built with Qml and Python, it may take a little more work to port it to Android.

Despite being written primarily in JavaScript, *Cinny* only has a web app and a desktop client. This still means that using it on a

smartphone should be easy. You can even add it using *Chrome* or with specialised apps such as *Hermit*.

Because *Fractal* uses the Gnome environment, it suits Gnome users well, but if you want to spread your wings, there may be a problem – unless you have a Linux phone.

NeoChat is likewise built using KDE in particular, which is difficult to support outside a regular desktop environment. With that said, it works perfectly fine on a Gnome desktop.

The choice of application also applies if you want to help the project, so pick carefully if that applies to you.

VERDICT

ELEMENT	9/10	NEOCHAT	8/10
CINNY	7/10	FRACTAL	7/10
MIRAGE	6/10		

Element comes out on top in terms of the number of platforms for which it has clients, enabling you to use it almost anywhere.

Encryption options

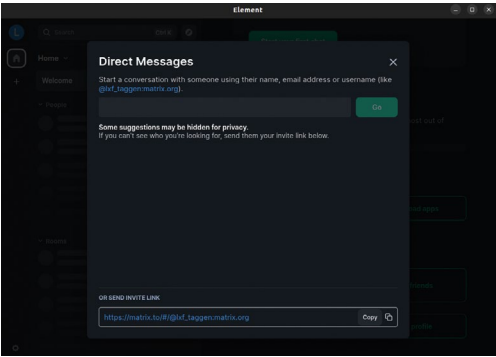
Look out for ways to protect privacy.

Before you see the full potential of Matrix, the clients probably look like any old chat box. But when you discover the more intricate details, you'll see the power of encryption and private rooms. The specification (<https://spec.matrix.org>) is open, clear of patent licensing and mostly uses HTTP. Importantly, it has vast support for encryption and is built for environments that anyone can join – so-called federations.

Any user should be able to create their own rooms and servers, making it vital that you know how to handle your encryption keys. Fortunately, all clients that support it have simple ways of doing this.

In this case, *Mirage* is disappointing because it does not support encrypted rooms or client verification. That said, *Fractal* and *NeoChat* also lack these essential features so cannot use Matrix to its full potential. You can still create private rooms but since encryption is not supported, you end up with a low-security environment. *Cinny* does support these vital features.

One useful feature is to have a web app available that can run in a browser. This makes it possible to have an application using



You might be looking for easy ways to send direct messages and encrypt them for your privacy.

standard tools. *Element* defaults to being a web app and can be a client for most platforms. *Cinny* also has such a web app, making it easy to use from a browser.

NeoChat and *Fractal* have no web app capability, because they are made for their respective desktop environments.

Other useful features would be to use different formatting for your messages and the overall look, which is where *Mirage* shines. Having a web app of *Mirage* would make it outshine the lot, but its real weakness is sadly encryption.

VERDICT			
ELEMENT	9/10	NEOCHAT	5/10
CINNY	5/10	FRACTAL	5/10
MIRAGE	7/10		
A lack of encryption options can let a chat client down.			

Frequency of updates

To use it, you need updates.

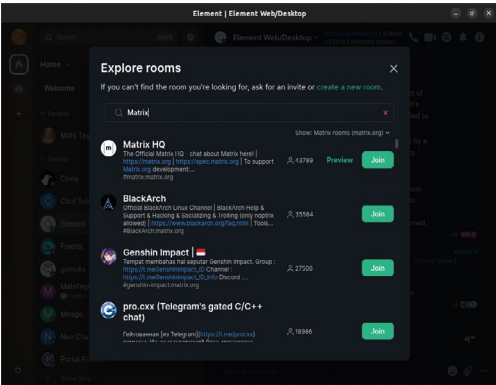
Here we're considering whether the client you pick will keep going for a reasonable time. If not, you should think twice before using it because you may end up with an unsupported product. To make a decision in this regard, you need to see how much development is going on and whether there are groups of people making it popular in different ways.

You can see on sites like GitHub whether any updates have happened lately and how much activity of other kinds is going on. If you look for support requests, you can also see how many people are using it. The more users, the bigger the incentive to continue developing the application. It is not always obvious by looking at just one of these indicators but combine them and you have a clear picture.

The winning client here is *Element*, which is supported by an entire company selling services for Matrix. This is a group that is completely dedicated to Matrix and its own client, making it a reliable choice when it comes to maintenance.

In stark contrast, *Mirage* has many fans but even the development team has not made much noise in the last 18 months. Without activity from these groups, all you can hope for is that your contribution can somehow get things running again. Sad, considering how pretty the application is.

The two clients that occupy the middle ground are *Fractal* and *NeoChat*, which are both supported by the bigger desktop



You can search for rooms for your issues but it takes more effort to know how well a client is doing.

groups. This may mean that there is a higher probability that someone will pick them up again and create newer, greater versions. You can also see that there is a new version of *Fractal* being developed as this *Roundup* is being written.

That leaves *Cinny*. This project has a promising 36 contributors on GitHub and keeps sending updates to the code. With several sponsors and such an active project, this client sees a bright future ahead. Add a few cosmetic features and it could be a real contender.

VERDICT			
ELEMENT	9/10	NEOCHAT	8/10
CINNY	7/10	FRACTAL	8/10
MIRAGE	4/10		
Element tops the table again in terms of updates and maintenance.			

The Verdict

Matrix chat clients

If you're choosing Matrix for your social needs, you are either a geek or you are tired of the flood of messages that have nothing to do with your query. You could be both, of course. In either case, switching takes something of an attitude adjustment. That attitude needs to be that every conversation is on point and does not wander from the subject matter.

With Matrix clients, you always create a room for your subject. A new room for each subject – keep it clean and stay civilised. In contrast, anything goes on Twitter and you have to filter out all the nutters.

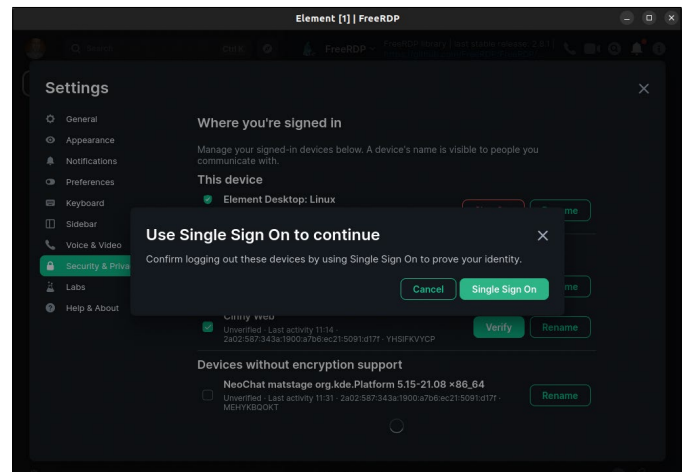
The other difference with Matrix is creating a private room, which is fully end-to-end encrypted, and is the default option. Having your own room and encryption keys requires a different way of thinking from simply throwing your ideas out into the current cyberspace black hole of traditional social networks.

All the packages in this *Roundup* have their advantages and disadvantages, but they are all usable. *NeoChat* and *Fractal* follow the standards of their respective projects but lack some vital features: end-to-end encryption and the verification of devices. *Mirage* suffers from a similar problem but has the looks down to a tee. It does have one more issue: there is no menu for settings; you have to tweak it with configuration files.

Element has all the settings you'd expect from a highly polished application and it also has all the features you could wish for. With one exception: you can't change its appearance very much.

You can expect future developments to offer audio and video communications. In some clients, they already exist but are in the early stages of development. Look out for fully fledged tools for doing that. Potentially, you can also look forward to different kinds of meeting rooms, which may require a central point for performance reasons. Sending lots of video feeds to everybody is a nightmare, and solutions could combine techniques around WebRTC to accomplish these things.

There are many solutions nowadays for fully private chat functions, but if you switch to Matrix, you have a standard that enables you to choose a software client without switching your identity.



1st **Element**

9/10

Web: <https://element.io> **Licence:** Apache 2.0

Version: 1.11.14 **OLM version:** 3.2.12

Element has a clear interface and is supported by serious people.

2nd **Mirage**

7/10

Web: <https://github.com/mirukana/mirage>

Licence: LGPL 3.0 **Version:** 0.7.2

Mirage looks polished and has potential but has stalled in development.

3rd **Cinny**

7/10

Web: <https://cinny.in> **Licence:** MIT

Version: 2.2.2

Cinny has the features but is harder to configure. Development is active.

4th **NeoChat**

7/10

Web: <https://apps.kde.org/neochat>

Licence: GPL 3.0 **Version:** 223.09

NeoChat is ready – if you love KDE, you will love its looks.

5th **Fractal**

6/10

Web: <https://wiki.gnome.org/Apps/Fractal>

Licence: GPL-3.0 **Version:** 4.4.0

Fractal is for Gnome users. It has a clear interface but nothing fancy.

» ALSO CONSIDER

Syphon is an up and coming client that is very much in early development. This makes it inaccessible for ordinary users but it could burst on to the scene at any point. When you download it, you end up with a directory containing the executable, which when run provides a barebones application that works but keeps throwing odd errors.

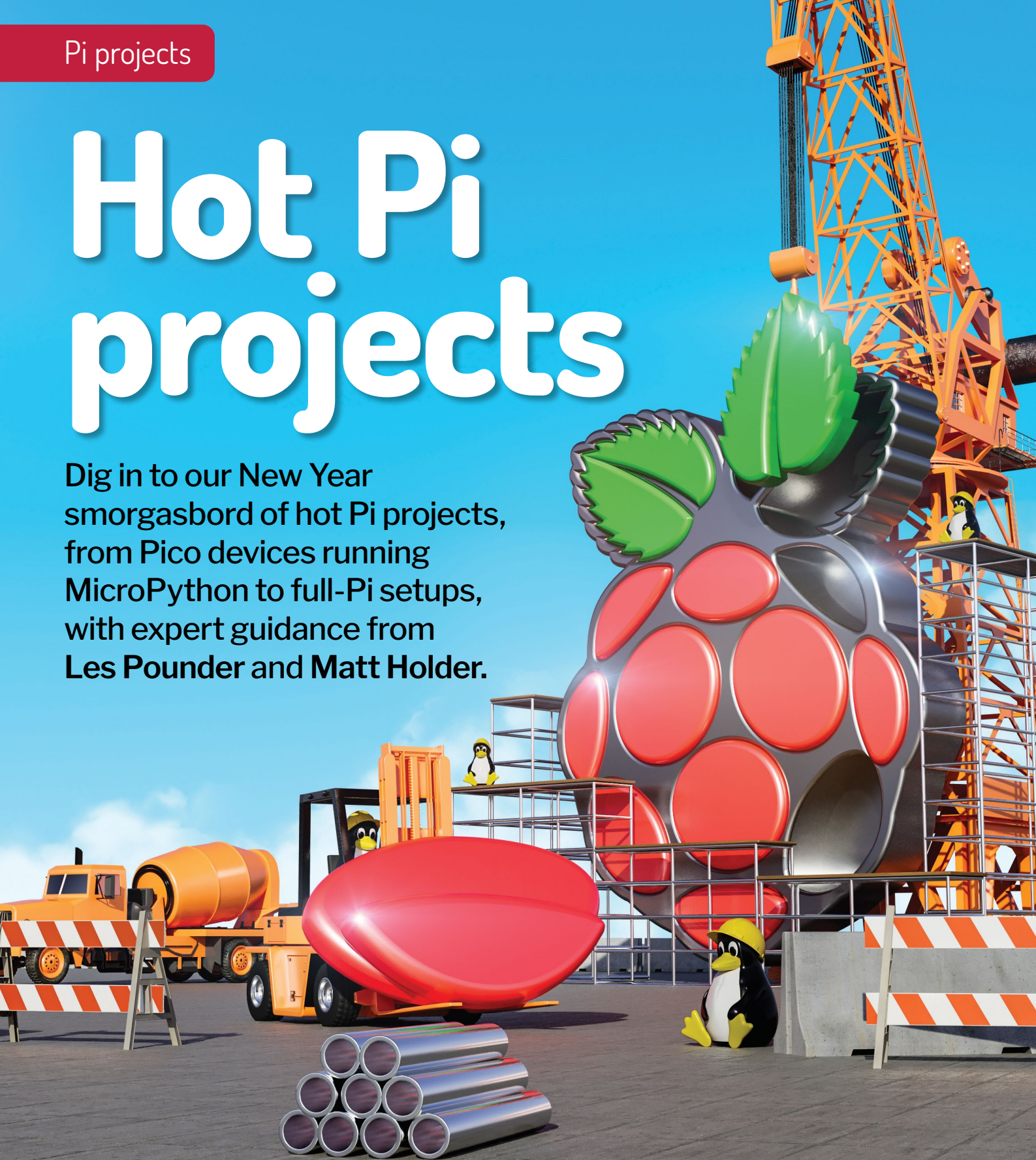
This is interesting, especially if you want to explore how things work and learn coding. In the long run, it is an

interesting alternative that's building for all platforms from day one. It comes with desktop, Android and iOS clients.

At this stage, it's mostly aspiring developers who will be interested. Anyone who is curious may want to know that Flutter is used to put it on screen. This gives you a wide array of systems you can install it on without running around a myriad of software development kits. The project is also very active on GitHub. **LXF**

Hot Pi projects

Dig in to our New Year smorgasbord of hot Pi projects, from Pico devices running MicroPython to full-Pi setups, with expert guidance from Les Pounder and Matt Holder.



The problem with owning a Raspberry Pi is knowing what to do with the flipping thing. That's if you've been able to get hold of one over the last year. But that is where the Pi Pico comes into its own, costing a fraction of the latest Pi 4 models and actually being available.

No matter what Pi you have to hand or what your ability is, we've got projects to keep you busy and out of trouble. Well,

mostly. We've selected self-contained capsule projects and have supplied the relevant code, while any kit you need shouldn't be too outlandish, from basic NeoPixels to simple radio receivers.

That's half the fun of Raspberry Pi projects: one moment you're flashing LEDs in different colours, the next you're reliving retro gaming moments, and then you could be up to your elbows in Python code. That last one is definitely fun – just

ask Les Pounder about trying to get his robot to go in a straight line...

The Pi's launch was revolutionary – it put a generous amount of computing power into the hands of enthusiasts for a ridiculously low price, all powered by open source. And it's still revolutionary, thanks to its investment in education, software, hardware improvements and, most recently, manufacturing its own silicon with the Pi Pico. So get making!

I like to make it, make it

Welcome to the maker revolution – but first, get your Pi up to speed.

To become a maker, you need the right kit, and without a microSD card, the main Raspberry Pi models won't even boot. So, to kick things off as quickly as possible, we've had our friends at Tom's Hardware do a bit of testing and come up with the fastest microSD cards to power your Pi.

Now is an amazing time to buy a fresh microSD card as prices have tumbled and speeds have increased dramatically. Previously, we would have recommended 32GB cards, but it's getting hard to find a 64GB card that costs much more than £7, while 128GB cards are regularly under £15. Much of this is powered by even larger capacity cards entering the market.

Theoretical speeds can top 140MB/s, with many advertising a healthy 100MB/s, as the old Speed Class measure tops out at Class 10 (10MB/s) there are now three new speed markings you now should look for:

UHS: Denoted with a U symbol and a number 1, 2 or 3 denoting 10MB/s, 20MB/s and 30MB/s respectively.

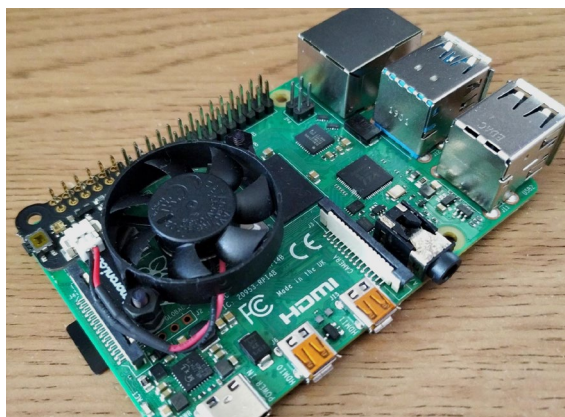
Video Class: Marked with a V and a number that denotes the MB/s, starting at V30, V60 and then V90.

App Class: The newest measure, which switches to IOPS (input/output operations per second), is denoted by A1, A2 and A3, although people have found Pi performance is less affected by IOPS.

Best for booting: All the big-name brands boot a Pi within 2.5 seconds of each other. The Kingston Canvas React came out first at 24 seconds, while the Samsung Evo Plus and SanDisk Extreme snuck in under 25 seconds, with the Samsung Pro Endurance and SanDisk Extreme Pro at just over 25 seconds.

Best for apps: All the main cards on test launched GIMP within 0.8 seconds of each other. The lesser-known Silicon Power 3D NAND being the fastest at 6.5 seconds, the Samsung Evo Plus, SanDisk Extreme Pro and Samsung Pro Endurance are just under 7 seconds, and the SanDisk Extreme at 7 seconds.

Fastest writes: Depending on your use case fast writes could be more useful than general performance. The



A Raspberry Pi 4 with Pimoroni's Fan Shim providing active cooling.

SanDisk Extreme Pro showed a clear lead on the rest at 38MB/s, the SanDisk Extreme came second with 32MB/s, and the Kingston Canvas React at 27MB/s. The Samsung Pro Endurance posted 21MB/s and the Silicon Power 3D Nand was just shy of 19MB/s.

Now you can boot your Pi as quickly as possible, turn the page to find a range of hot projects. We also recommend installing your Pi in a case – we've listed the best to consider at the end of the feature.

Raspberry Pi	Config.txt
Pi Zero (W)	arm_freq=1100 core_freq=450 over_voltage=6
3A+/B+/CM3+	arm_freq=1500 over_voltage=4
Pi 3/CM3 & Pi 2 v1.2	arm_freq=1300 core_freq=500 over_voltage=4
Pi 2 v1.1	arm_freq=1000 core_freq=500 over_voltage=2

» OVERCLOCK YOUR PI

It's easy to overclock your Pi to gain a modest speed increase. If you're serious, first attach a suitable Pi cooling fan, such as the Pimoroni Fan Shim.

It's worth a `sudo apt update && sudo apt upgrade -y` then open **config.txt** for editing (in the **/boot** directory):

```
$ sudo nano /boot/config.txt
```

At the bottom of the file for a Pi 4, 400 or CM4, add these new lines – see the table (above-right) for other models:

```
arm_freq=2200
```

```
gpu_freq=750
over_voltage=6
force_turbo=1
```

If you choose to use the same SD card in a different model of Raspberry Pi later, you should change or remove these lines. Save the file by pressing Ctrl+X, Y then Enter.

Reboot the Raspberry Pi. If it fails to boot, power off the Raspberry Pi, remove the microSD card, edit **config.txt** and check your settings. If they're

correct, reduce the clock by 100MHz, check your cooling and try again.

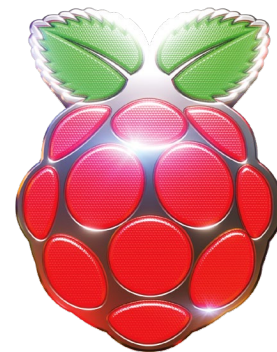
With the Pi booted to the desktop, open a new terminal. To see its current CPU speed updated every second, use:

```
$ watch -n 1 vcgencmd measure_clock arm
```

When idle, the CPU runs at around 600MHz. If we were to open a web browser or another heavy application, the CPU speed would increase to match the demand.



Play retro games



Defend the universe, save the princess, win the race and all without spending a 10-pence piece!

YOU NEED

- > **Raspberry Pi 3, 4 or Zero 2W**
- > **8GB or larger microSD card**
- > **Controller (USB or Bluetooth)**

In the '80s, the only way to experience arcade-quality gaming was by going to an arcade. Over the passing decades, arcades have become mere memories and gaming is now a home pastime. For the more 'mature' gamer, the appeal of retro gaming is nostalgia, and the Raspberry Pi has been a capable retro gaming platform for most of its life. In this tutorial, we're converting a Raspberry Pi into a retro gaming console using *RetroPie*.

Before we get too far, a word on legality. Retro games are still under their respective owners' copyright. In some countries this isn't an issue, in others it is. Please consult your local regulations.

RetroPie is a great way to play classic retro games via a Pi. We should know, as we have a Picade cabinet on our bookshelf that runs a customised version of *RetroPie*. Installation is simple – download and install *Raspberry Pi Imager* from www.raspberrypi.com/software/ and open the app. In the Operating System menu, select Emulation and Game OS, then select *RetroPie* for your model of Pi. Insert an 8GB or larger microSD card and select it from the Storage menu. Click Write to download and write the image to the card. Remove the card and boot your Pi from it.

On the first boot, we need to set up a controller. This needs to be a controller physically connected to the Pi via USB. Bluetooth doesn't work at this time. Follow the instructions and map all the relevant controls. Once done, we see the *RetroPie* interface, EmulationStation.

Games are collected by systems, and without any games, no systems show in the menu. Your legally owned ROMs can be transferred to *RetroPie* using a



The *RetroPie* user interface is made for controllers. Games are broken down into their respective console or computer for easy reference.

FAT32-formatted USB drive. Create a folder, **RetroPie**, on the USB drive and insert it into the Raspberry Pi running *RetroPie*. Leave the drive for five minutes for *RetroPie* to create a folder structure for the ROMs. Remove the USB drive, place it into another computer and copy your ROMs into the correct folder under **/retroPie/roms/**. Note that the number of ROMs should not exceed the capacity of the *RetroPie* microSD card. Remove the USB drive and place it back into the Pi. The ROMs auto-copy to their respective folders on the Pi. If you're copying gigabytes of ROMs, this takes a long time; if it's only a few hundred MBs, it should be done within a few minutes. When done, remove the USB stick and reboot *RetroPie* (Start > Quit > Restart). Some systems, such as Arcade, MAME and PlayStation, require BIOS images.

Controllers

To use a Bluetooth controller, go to the *RetroPie* menu. Scroll down and select Bluetooth. Select Pair And Connect Bluetooth Device. The name and MAC address of the controller should appear (it is normally printed on the controller) – make sure they match. To validate the connection, try the first security mode; if that fails, try the next. Once done, your controller can be used from the comfort of your sofa.

On the Raspberry Pi 4, the best level of emulation we can achieve is Nintendo 64. Even then, some games, such as the seminal *Goldeneye 007*, may run a little choppy. PlayStation 1 and some 2D PS2 games should run rather well.

We've only just got you started with *RetroPie*. For more information, head over to <https://retropie.org.uk> and learn how to get even more retro gaming fun from your Raspberry Pi.

QUICK TIP

RetroPie can be heavily customised, under the *RetroPie* menu – look for ES Themes and Splash Screens. The former can change the UI, the latter can change the boot splash for a bespoke cabinet.

Sega's MegaDrive (Genesis in the US) is a classic 16-bit console, and something that all models of Raspberry Pi can emulate.



CREDIT: Sega

Make a media centre

Settle down with your own open source media centre and watch the latest blockbusters with a side of Pi and popcorn.

YOU NEED

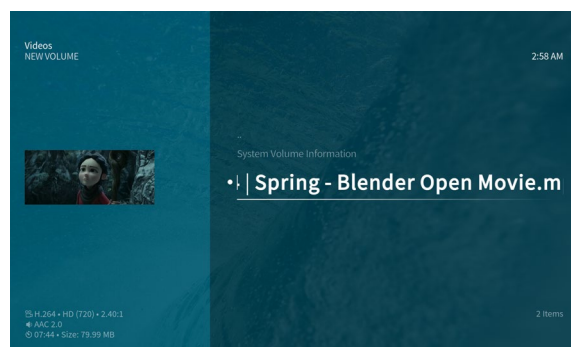
> **Raspberry Pi 3, 4 or Zero 2W**
> **8GB or larger microSD card**

When the Raspberry Pi first arrived in 2012, one of the first projects we saw was a media centre. Even the humble Raspberry Pi 1, with just a single 700MHz CPU and 512MB RAM, was capable of decent 1080p video playback. Fast-forward to today and the Raspberry Pi 4 is many times more powerful and able to perform much better with our growing library of media. In this tutorial, we explain how to install a media centre OS to your Raspberry Pi, and then show you how to play media from USB and live streams, all from the comfort of your sofa.

OSMC is a great introduction to turning your Raspberry Pi into a media player. This lightweight Linux OS is designed just for serving media, be it local or streaming video, or music. Installation is simple: just download and install *Raspberry Pi Imager* from <https://www.raspberrypi.com/software/> and open the app. In the Operating System menu, select Media Player OS and then select OSMC for your model of Pi. Insert an 8GB or larger microSD card, then select the card from the Storage menu. Click Write to download and write the image to the microSD card. Remove the card and boot your Raspberry Pi from it.

On the first boot, we need to set the language and our location. Once done, we are presented with a simple UI that is made for media consumption. Media is categorised into sections (Videos, Music, Pictures) and clicking on these headings opens additional menus. Before that, we need to go into Settings > My OSMC and from there select the top icon: Network. In the Network menu, use the arrow keys (or mouse) to select Wireless and then enable the adapter. Select your Wi-Fi SSID and connect. When done, press Esc or Backspace until you reach the main OSMC menu.

To play local media, such as video files on a USB stick, insert the USB drive in the Raspberry Pi and go to Videos, press Enter and you will see that the drive is auto-mounted. Select the drive and navigate to the video file. Note that video files need to be in a format compatible with the Raspberry Pi – usually MP4 or MKV. If not in this format, use the free *Handbrake* tool to transcode them on a more powerful PC. Press Enter to play your video. The on-screen UI can be toggled on and off by pressing the right mouse button. To exit, press Esc, which returns you to the main menu. The Audio menu



This simple interface is made for the big screen and from it we can devour our media libraries and watch streaming media.

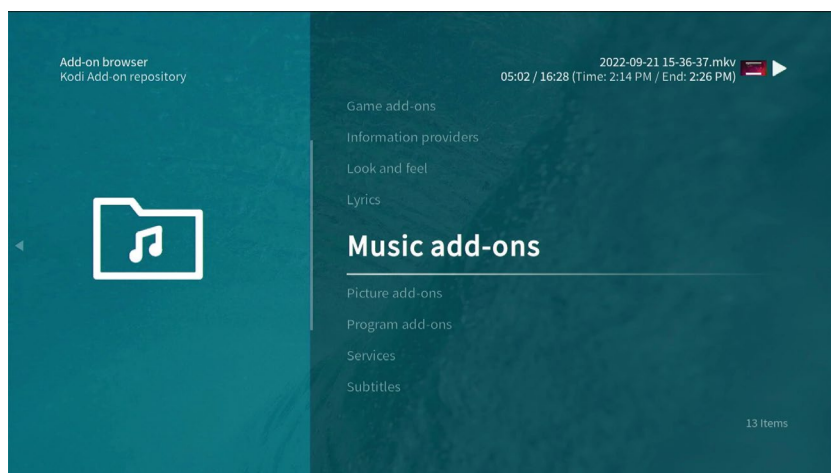
behaves in the exact same manner, except that it can play audio files in many different formats.

Streaming media such as YouTube can be easily accessed via OSMC. Go to Settings > Add-On Browser and select Install From Repository. As we have already connected the Pi to the internet, a list of add-on categories appears. Select Video Add-Ons and press Enter. A new list of compatible add-ons is presented. We chose NASA's live streams. Scroll down to NASA and install. This triggers the add-on to request the installation of YouTube, as this is how NASA serves its videos. With installation complete, go back to the main OSMC menu. Select Videos, then Video Add-Ons. NASA is now a selectable menu with the latest streams. Using the same process, we can also install music and game add-ons, such as a MAME arcade machine emulator – but instead of doing that, you can check out the retro gaming tutorial (*opposite page*).

For more information about OSMC, head over to <https://osmc.tv> and discover how to get the very best from this great media operating system.

QUICK TIP

Infrared remote controls can be used with the Raspberry Pi – just search for FLIRC. Wireless keyboards with trackpads make excellent controllers.



Adding extra features, such as streaming services, is made possible via the add-on repository.

Build a Wi-Fi router

Create a Raspberry Pi-based Wi-Fi router as a secure option for use around the home.

YOU NEED

- > **Raspberry Pi 4 Model B (2GB is ample)**
- > **16GB or 32GB microSD card**
- > **Suitable PSU**

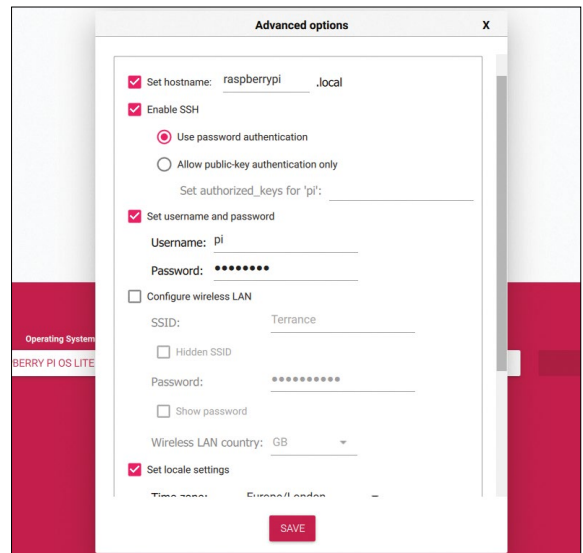
With the release of the Raspberry Pi 4, the networking stack was greatly improved thanks to changes to the hardware, which meant that USB bus was no longer used to make it available to the operating system. In this project, we are creating our own Wi-Fi router using Raspberry Pi OS and *RaspAP*. Once the initial setup has been completed, the wireless router can be configured via a web interface – much like the products available from popular router manufacturers.

What's the benefit of making this yourself? Well, the first reason is that cheaper routers aren't kept updated for very long, possibly leaving security holes. Basing the router on Debian means that the base operating system is incredibly well tested and security updates are released on a regular basis. The web GUI also enables traffic usage graphs to be viewed, connections made to OpenVPN and WireGuard tunnels, the use of ad-blocking DNS services and much more. But enough introduction – let's begin!

The project is completed in three stages. Stage one is to prepare the microSD card with the Raspberry Pi OS image and enable SSH access from first boot. Stage two is to install the *RaspAP* software, while the final stage is to configure the router and connect devices to it.

To prepare the image, first of all follow the instructions at the following web address to install the *Raspberry Pi Imager* tool: www.raspberrypi.com/software/. Once installed, open it from the launcher and select Raspberry Pi OS (Other) > Raspberry Pi OS Lite (64bit) from the Operating System menu. Next, from the Storage menu, select the relevant microSD card (if multiple options appear in the list, make sure you don't pick the wrong one because you'll lose whatever is stored on it). Click on the settings cog and select the options to enable SSH with password authentication, set the username and password, and set the locale to the correct time zone and keyboard layout. Click on OK to confirm the options and then click on the Write button. Wait while the image is being downloaded and written to the microSD card.

When the image has been written to the microSD card, remove it from your device and plug into the Raspberry Pi, and connect the Pi to the power supply and wired network. The lights then flicker as the device boots up. Once booted, you can SSH to the Raspberry Pi from the terminal using `ssh <USERNAME>@IP ADDRESS`. Refer to the DHCP leases on your router if the IP address is proving tricky to find. Enter the following commands to make sure the system is fully updated and that the correct region is set for the Wi-Fi configuration – navigate through the *raspi-config* tool to find the correct settings:



■ Use Raspberry Pi Imager to write the image to a microSD card.

```
$ sudo apt update
$ sudo apt full-upgrade
$ sudo raspi-config
$ curl -sL https://install.raspap.com | bash
$ sudo reboot
```

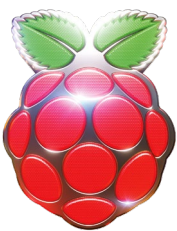
Once you have rebooted the Raspberry Pi, the initial details need to be used to connect a device to the router. These details can then be changed from the web GUI.

```
IP address: 10.3.141.1
Username: admin
Password: secret
DHCP range: 10.3.141.50 — 10.3.141.255
SSID: raspi-webgui
Password: ChangeMe
```

Once connected, open a web browser and navigate to **10.3.141.1**. Log in with the details above, then navigate to Authentication to change the login password and go to Hotspot > Security to change the details of the SSID being connected to. When we first connected to the new AP, a DHCP address was granted, but it was not possible to browse the web. To cure the issue, first of all edit the `/etc/sysctl.conf` file as administrator and uncomment the line that reads `net.ipv4.ip_forward=1`. Save the file and then run the following commands before rebooting the Raspberry Pi:

```
$ sudo iptables -t nat -A POSTROUTING -j MASQUERADE
$ sudo iptables-save | sudo tee /etc/iptables/rules.v4
```

In this article, we have only had space to cover the very basics of how to get this tool up and running, so take your time and discover more about the software.



Track and scan planes

Use your Raspberry Pi to track aircraft or detect local devices to keep an eye on all kinds of data.

YOU NEED

- > **Raspberry Pi 4 Model B (2GB)**
- > **16GB or 32GB microSD card**
- > **5V PSU**
- > **ADS-B receiver**
- > **Realtek RTL2832U + R820T2**
- > **USB extension cable**

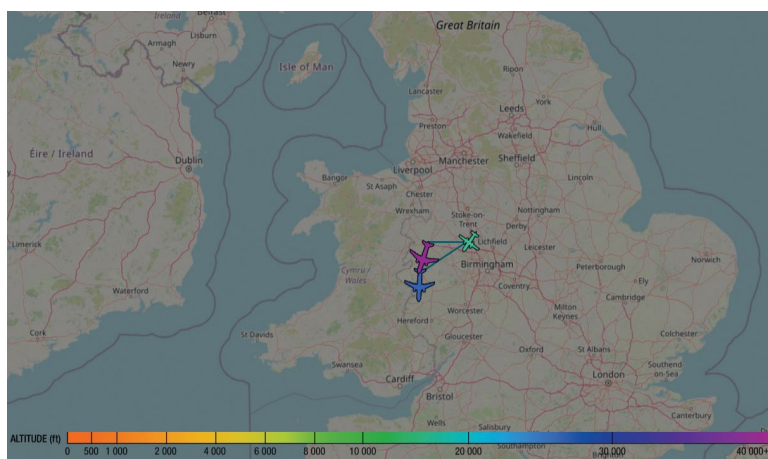
Over the last few years, there has been much talk of software-defined radio (SDR – see **LXF273**). This is a way for a generic piece of receiver hardware to be repurposed to scan different frequencies or be used for a whole variety of purposes. One example of this is the Realtek RTL2832U+R820T2-based USB stick (around £35), which was originally intended to be used as a TV receiver. For this project, we will briefly introduce how this receiver can be used to receive ADS-B signals from passing aircraft and how this information can be fed into a web service that aggregates signals from many users, from across the globe. The second thing we will discuss is how to use the same receiver to scan for any local devices that send data at 433MHz or 868MHz.

Plane tracking

ADS-B Exchange is a service that aggregates data from a huge number of receivers, which are situated across the globe. The project styles itself as the world's largest source of unfiltered flight data. Some other services only display certain types of flights – for example, they exclude military aircraft. To install the project's image, utilise the *Raspberry Pi Imager* to install a custom image, which can be downloaded from the following link: www.adsbexchange.com/how-to-feed/adsbx-custom-pi-image.

Once written, the card can be put into your Raspberry Pi, the receiver can be connected via USB and it can then be connected to the power. After booting, connect to the ADSBx-config Wi-Fi SSID and, from a browser, visit <http://adsbexchange.local>. Click on Wi-Fi Setup and add the details for your home network. Get reconnected to your home network, visit the same address again and select Configure Receiver/Location. From here, add your longitude and latitude, and select the Save And Reboot option.

A reboot later, connect to the same address again and explore the interface, which shows any detected planes or helicopters. If you wish, you can enable the Use Custom LED Indications setting, which repurposes the Pi's LEDs to give you an idea of what is currently being detected. To determine whether your own instance is reporting data back to ADSB-Exchange, you can visit <http://adsbexchange.com/myip> – this queries the database, using your internet-facing IP address. The initial password for the Pi user is **adsb123** and this



Live aircraft tracking data from our Raspberry Pi. Elon, is that you?

should be changed using the Change Password option from the web GUI. Once this has been changed, access via SSH and for the web GUI uses your new password. The local interface can display graphs of detected aircraft and a map display. Viewing from the following address displays aggregated data: <https://globe.adsbexchange.com>.

Scanning at home

When preparing this tutorial, we were able to detect and interpret the data from an Oregon weather station. This data could then be used to feed online services or sent, via MQTT, to a local home automation system.

To set up the relevant software, first of all prepare a new installation of Raspberry Pi OS using the instructions on the build your own Wi-Fi router project (see *opposite page*). Once this has been completed, connect the receiver to a free USB port and install the necessary software:

```
$ sudo apt install rtl-433
```

Once installed, the following two commands can be used to scan at either 433MHz or 868MHz:

```
$ sudo rtl_433 -C si
```

```
$ sudo rtl_433 -f 868M -s 1024k
```

Running these commands for a few minutes each determines what devices are transmitting on these frequencies and what useful information can be utilised from them. While running the commands, look for any old remote-controlled sockets, remotes and devices of that nature that could be utilised. Data returned can be reformatted into JSON and piped to an MQTT server, for further usage:

```
$ rtl_433 -F json -M utc | mosquitto_pub -t  
home/rtl_433-l
```

Further information about this useful tool can be found at https://github.com/merbanan/rtl_433.



Master the Pi Pico GPIO

Get to grips with the Raspberry Pi Pico's GPIO to open the gateway to a range of simple projects.

YOU NEED

- > **Raspberry Pi Pico or Pico W**
- > **A push button**
- > **An LED**
- > **330 Ohm resistor (orange-orange-brown-gold)**
- > **A breadboard**
- > **Code:**
<https://github.com/leesp/LXF-Pi-Special/archive/refs/heads/main.zip>
- > **SD card**

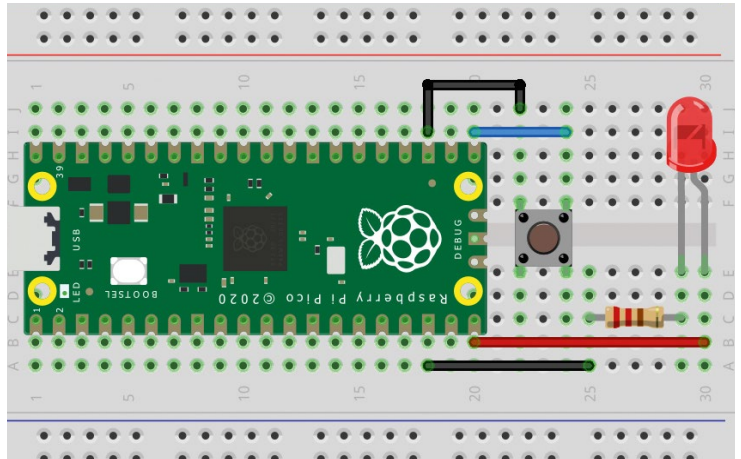
The Pi Pico comes pre-installed with MicroPython, a version of Python 3 for microcontrollers. It is regularly updated and new versions can be found at <https://bit.ly/lxf298upython>. It is prudent to update MicroPython before trying this tutorial.

Our circuit for this project is a simple LED toggle switch – a detailed circuit diagram is included in the download link. A push button is used to turn the LED on and off. The Pico reads the GPIO pins and reacts to the change. Download and install *Thonny* for your OS and connect your Pico to the computer. Open *Thonny*, go to Tools > Options and click on the Interpreter tab. Set the interpreter to be MicroPython (Raspberry Pi Pico), then set the serial/COM port to auto-detect the Pico. *Thonny* should now detect the Pico and the Python shell confirms this.

We start the project code in a blank document. The first step is to import two modules of prewritten code that will enable us to interact with the GPIO pins, and control pauses or delays in the code:

```
from machine import Pin
import time
```

Next we create two objects. The first, **led**, sets up our LED on GPIO15 as an output – we want current to flow to it in order for it to illuminate. The second object, **button**, is a push button on GPIO16. This is an input and we pull the pin high so it has current flowing. When the button is pressed, it connects GPIO16 to GND, causing the pin to pull low and toggle the LED on/off:



Your first electronics project on the Raspberry Pi Pico is intentionally easy. It builds our knowledge and confidence to tackle the next project.

```
led = Pin(15, Pin.OUT)
button = Pin(16, Pin.IN, Pin.PULL_UP)
```

Turn the LED pin off. This is basic housekeeping – the pin should already be off.

```
led.low()
```

Create a **while True** loop that will indefinitely run the project code:

```
while True:
```

The code is now indented inside the loop; this means that it works within the loop. The first task is to create a conditional test that checks whether the button has been pressed. If it has been pressed, the GPIO pin changes state from high to low (0).

```
if button.value() == 0:
```

When the button is pressed, the LED is toggled on or off, depending on its current state. A short message is printed to the Python shell for debug purposes. A delay of 0.2 seconds is added in order to reduce accidental double pushes, called debounce:

```
led.toggle()
print("Toggle")
time.sleep(0.2)
```

The final two lines are part of the conditional test and relate to when the button is not pressed. In this circumstance, **pass** is used to allow the loop to repeat:

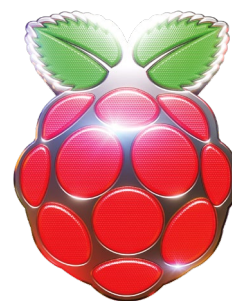
```
else:
    pass
```

Click on File > Save and save the code as **button_blink.py** to the Raspberry Pi Pico. Click on the run button (green play button) to start the code. Now press the button to toggle the LED on and off.

This introduction is just a fraction of what the Raspberry Pi Pico can do. For more projects, head to Tom's Hardware, where there is a full section of projects to try out: <https://bit.ly/lxf298toms>.

» THE RASPBERRY Pi PICO

The Raspberry Pi Pico arrived in early 2021 and saw the first Pi silicon device hit the market, the RP2040. The RP2040 system on chip (SoC) is an Arm Cortex M0+ dual-core at 133MHz (which can be easily overclocked), with 264KB of SRAM and 2MB of onboard flash storage. This may not sound like a lot, but in the world of microcontrollers, it is plenty. The Pico has a 40-pin GPIO, just like the larger Raspberry Pi, but these pins are located around the perimeter of the board and make for easy insertion into a breadboard. The pins do not come presoldered and feature castellated edges. This means we can either solder male header pins to the board, or use the castellations to surface-mount solder the Pico to a circuit of our own design. The GPIO is easy to use via MicroPython, CircuitPython, JavaScript, C, Arduino and many other languages. The Pico also has the power to easily emulate '80s-era home computers, such as the ZX Spectrum and Commodore 64. Not bad for a £4 device.



Convert currency

Work with remote data using an API and the Pi Pico W to make a live currency converter.

YOU NEED

> **Raspberry Pi Pico W**
 > **Code:**
<https://github.com/lesp/LXF-Pi-Special/archive/refs/heads/main.zip>

QUICK TIP

JSON (JavaScript Object Notation) is convenient and easy to use, but large objects can become jumbles of noise. Use an online viewer (<http://jsonviewer.stack.hu>) to explore the data and find the correct keys.

In this project, we're creating a simple app to get the latest exchange rates for the euro and US dollar against the UK pound. The Raspberry Pi Pico W comes with MicroPython, a version of Python 3 for microcontrollers. It is regularly updated and new versions can be found at <https://bit.ly/lxf298upython>. It's a good idea to update MicroPython and ensure you get the version for the Pico W, as it has extra modules for Wi-Fi connectivity.

Download and install *Thonny* for your OS and connect your Pico to the computer. Open *Thonny*, go to Tools > Options and click on the Interpreter tab. Set the interpreter to be MicroPython (Raspberry Pi Pico) and then set the serial/COM port to auto-detect the Pico W. *Thonny* should now detect the Pico W and the Python shell should confirm this.

Start the project code in a blank document. The first task is to import libraries of prewritten code – modules, in Python parlance. These modules enable our Pico W to connect to a network and make requests to remote machines over the network:

```
import network
import urequests
```

Next create an object, **wlan**. This object represents a connection between the code and the Pico W's Wi-Fi chip. You need to set the connection to active to enable the Wi-Fi chip:

```
wlan = network.WLAN(network.STA_IF)
wlan.active(True)
```

Connect the Pico W to your Wi-Fi access point. Change **YOUR SSID** and **YOUR PASSWORD** to match those of your router and keep the quotation marks. Print the connection status to the Python shell. **True** means we are connected, **False** we are not:

```
wlan.connect("YOUR SSID", "YOUR PASSWORD")
print(wlan.isconnected())
```

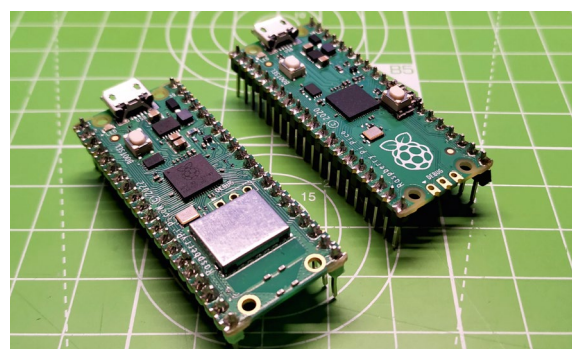
The exchange rate data is stored in an online resource that uses JSON, an open standard file format. By creating an object, **gbp**, we can make a network request for the data in JSON format by getting the data from the URL:

```
gbp = urequests.get("https://api.exchangerate-api.com/v4/latest/GBP").json()
```

Data is stored inside the **JSON** object as a series of keys and values. By giving the name of a key, **JSON** gives us the value. So, using **USD** as a key, we return the value of the pound to the US dollar and store it in a variable, **dollar**. The same applies to the value of the pound to the euro.

```
dollar = gbp["rates"]["USD"]
euro = gbp["rates"]["EUR"]
```

Print the exchange rate data and mix it into a sentence so it has context. Here we explain the current value of the pound to the US dollar and the euro. The



The only outward difference between the Pico and Pico W is the large silver square in which the Wi-Fi chip resides.

variables (**dollar** and **euro**) are appended to the end of the sentences. The comma is used to add a space:

```
print("The current value of the UK Pound to one US Dollar is £", dollar)
print("The current value of the UK Pound to one Euro is £", euro)
```

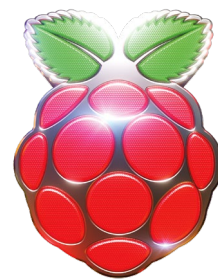
Click File > Save and save the code as **Pico_API.py** to the Pi Pico W. Click on the run button (green play button) to start the code. In the Python shell we see either **True** or **False** appear. **True** means we are connected to Wi-Fi. The next two lines we see are the current exchange rates for the dollar and euro. We could mix the code from the Raspberry Pi Pico GPIO tutorial (*opposite*) into this project to create a tool that fetches the latest data at a button press. That is the beauty and power of MicroPython and the Pi Pico W.

» PICO PRODUCTS

The Raspberry Pi Pico W has spawned a wave of products that embed the board as the centre of a project. Pimoroni, for instance, has created Pico W Aboard. This new range sees the Pico W surface-mount soldered, via its castellated pins, to a range of products, such as weather stations (Enviro Weather), RGB LEDs (Plasma Stick 2040 W) and the ludicrously named Galactic Unicorn, which has no bearing on intergalactic equines.

There are also sensor kits – Enviro Urban, Indoor and Grow – that can monitor air quality, temperature, air pressure and our plants. These kits come with an onboarding process, hosted via a micro web server on the Raspberry Pi Pico W – yes, you can run a tiny web server from the £6 Pico W! We made one in **LXF294**. Details about Pimoroni's Pico W Aboard products can be found at <https://shop.pimoroni.com>.





Build a smart home

Use ESPHome to feed a wide range of useful data to a smart home hub via a Raspberry Pi Pico W.

YOU NEED

- > Pi Pico W
- > DS18B20 temp sensor
- > 4.7kOhm resistor
- > Dupont jumper cables
- > MicroUSB power adaptor
- > An instance of Home Assistant (see LXF287)

ESPHome was originally written to allow the ESP8266 and ESP32 devices to easily interact with *Home Assistant*. Using YAML syntax, the required options and configuration can be written, then parsed and used to create source code, which is cross-compiled and uploaded to the devices. With the release of the 2022.11 version, developmental support has been added for Raspberry Pi Pico W boards.

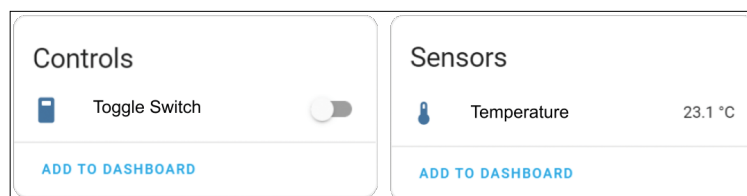
In this project we're creating a program that enables us to read the temperature from a DS18B20 sensor, report it to *Home Assistant* and flash the internal LED on the board via a switch in *Home Assistant*'s web GUI.

To begin, we need to set up *Home Assistant* and enable the ESPHome add-on. The easiest way to run the software is to install it on a microSD card, which is then inserted into a Raspberry Pi. If using this option, download the image from www.home-assistant.io/installation, follow the instructions in the build your own Wi-Fi router article (page 36), and select the custom image that has been downloaded. Alternatively, use the generic x86-64 image and install it as a VM on your laptop or PC. Once installed and running, follow the onboarding instructions to finish the installation: <https://bit.ly/lxf298onboard>. Once done, use the My Add-On link (<https://bit.ly/lxf298addon>), which makes installation of the add-on a lot easier. Once you are logged into *Home Assistant*, select the option to install the add-on, start it at boot and enable the watchdog. Start the add-on and select the Open Web UI link.

Before creating your device, select the Secrets link and add these options, then save and close the file:

```
wifi_ssid: ADD_SSID
wifi_password: ADD_PASSWORD
```

We will now create a new device and add some code to it. Click on New Device, then Continue and enter a name, which will be shown in the ESPHome dashboard. When asked to select the board type, click on Raspberry Pi Pico W and then select Next. Once created, select Skip and then click the Edit button on



The dashboard created by adding the integration can be seen here. Temperature information can now be used in automations and Home Assistant dashboards.

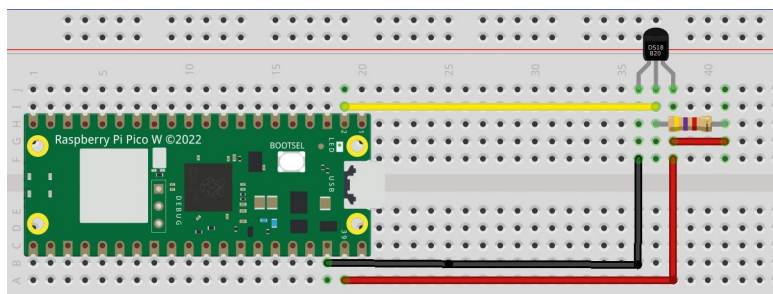
the new device. At the bottom of the file, add the following lines:

```
dallas:
  - pin: GPIO1
sensor:
  # - platform: dallas
  # address: ADD_HEXADDRESS
  # unit_of_measurement: "°C"
  # name: "Temperature"
output:
  - platform: gpio
    pin: 32
    id: 'output_component1'
light:
  - platform: binary
    name: "State LED"
    output: output_component1
```

Now save the file and click on the Install button. When asked, select manual download. Once the compilation has completed, download the file and connect the Raspberry Pi Pico W to your PC or laptop with the Bootsel button pressed down. Copy the downloaded **UF2** file to the new drive that has appeared on your device. When the file has copied and the device has rebooted, it connects to the Wi-Fi and then ESPHome. The device then shows as online and is available to be added to the *Home Assistant* dashboard. A notification is displayed on the sidebar and by selecting this and using the API password from the file we edited earlier, it is added. Note the commented lines in the file – these need to be uncommented and the address replaced once the Logs option has been selected from ESPHome and the address found from this page. When the sensor's address has been changed, the lines uncommented and the file saved again, click on Install and this time the Wirelessly option can be used.

Finally, from *Home Assistant*, navigate to the Settings > Devices & Services page and under the ESPHome option, select the name of your newly created device and then 1 Device. From this page, you can see the measured temperature, and by toggling the switch, the on-board LED switches on and off.

Use this circuit diagram to connect the temperature sensor.



Control NeoPixel LEDs

Create animated lighting effects with the brightest LEDs connected to your Raspberry Pi Pico.

YOU NEED

- > Any Pi Pico
- > Micro USB cable
- > Thonny Editor
- > MicroPython firmware
- > WS2812B LEDs (with suitable power)

We're using MicroPython here to control a small set of LEDs. Known as NeoPixels, they are a string of LEDs that are generally multi-coloured and can be addressed individually. NeoPixel is the name given to this style of LED by Adafruit, but the same driver is used for a large number of them (WS2811/WS2812), no matter who the manufacturer or supplier. These models are available with

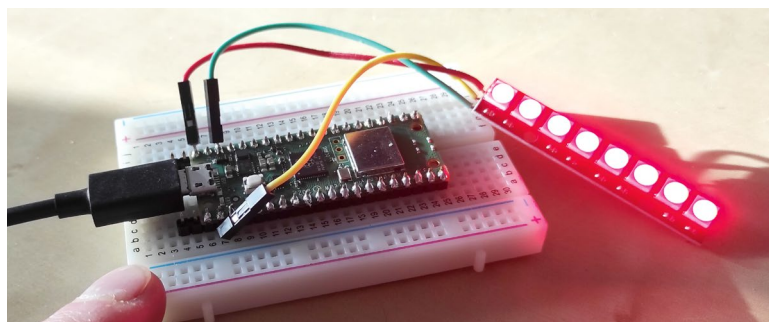
separate RGB channels, so that any colour can be made by mixing the values (think of HTML colours).

First, download and install MicroPython from <https://bit.ly/lxfupdate>. Then hold down the Bootsel button on the Pico board and connect to your PC or laptop. Copy the downloaded file to the newly mounted disk, then the Pico W reboots and loads MicroPython.

Download and install the *Thonny* IDE from <https://thonny.org>. Open the IDE and, from the bottom-right, select MicroPython (Raspberry Pi Pico). This connects the terminal in the bottom of the window to the Pi Pico's REPL (Read-Eval-Print Loop) – this allows code to be entered line by line. This is useful for basic tests, but not very useful when trying to write a program. Programs can be entered into the main section of the window, then saved to the board. When saving, you can choose between the local computer and the Pi Pico. When saving to the Pico, use the filename **Boot.py** and the code executes as soon as the board boots.

Step three is to connect the Raspberry Pi Pico or Pico W to the LEDs and the PC for programming (see the circuit diagram below).

The following code smoothly pulses the LEDs. They start with a dim red colour and gradually build to full



Here you can see the LEDs running the program we have written.

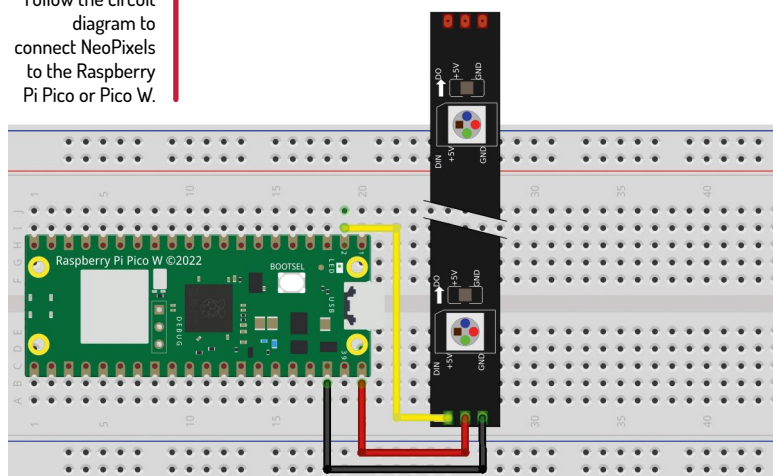
brightness before fading back down again to nothing, before this cycle repeats.

```
import machine
import neopixel
import time
#Pulse all LEDs
numPixels = 8
np = neopixel.NeoPixel(machine.Pin(1), numPixels)
pulseTime = 2
numLoops = 100
while True:
    for a in range(1,numLoops):
        for i in range(0,numPixels):
            np[i] = (int(a*254/numLoops),0,0)
            np.write()
            time.sleep(pulseTime/(2*numLoops))
        for a in range(numLoops,1,-1):
            for i in range(0,numPixels):
                np[i] = (int(a*254/numLoops),0,0)
                np.write()
                time.sleep(pulseTime/(2*numLoops))
```

In this code, we first import the necessary libraries, then store the number of pixels in a variable, create an object to reference and set values on the LEDs, then two more variables. The larger the value of **numLoops**, the smoother the effect. Next we define a **while** loop that will run forever, before setting a **for** loop that runs for the number of times defined in the **numLoops** variable. For each iteration of this loop, we set each LED of the string (using the inner loop) to a brightness, as defined by the value in the outer loop. After we have set the value of each LED, we write the values that cause the LEDs to change state, wait for a short period before iterating again and getting brighter each time. Once this outer loop has completed, the second stage, decreases the brightness until the LEDs are practically switched off. The colour can be altered by changing the numbers in the brackets when setting **np[i]**. The values in the tuple represent red, green and blue.

You can find two other examples of effects at <https://github.com/mattmole/LXF298>.

Follow the circuit diagram to connect NeoPixels to the Raspberry Pi Pico or Pico W.



We're on the case!

A case isn't just for Christmas – it's a life-long home for your chosen Raspberry Pi project, so make sure you pick a good one!

Although you can run a Raspberry Pi effectively without one (Raspberry Pi co-founder Pete Lomas designed it that way), many people want a case that can protect the single-board computer while offering an attractive design, built-in cooling or even special features. Originally, the Raspberry Pi was not supposed to be contained in anything, but over the years the number of cases available has become almost as numerous as decimal places in the number pi, so choosing the right one for your model and project can be a challenge.

There's no single best Raspberry Pi case for all uses, because what you would want for a media centre is different from what you need for maker projects or extreme overclocking. Some cases can turn the humble SBC into a desktop computer, while others are designed like classic retro consoles to hit the nostalgia node of our brains, and some cases completely engulf the Pi in a shield of aluminium or acrylic. Let's start by considering what you need from a case:

GPIO, camera, display

Many cases limit or completely block your access to the GPIO pins, which you need for attaching the best Raspberry Pi HATs and other lights, motors and sensors. Before you buy, consider whether you plan to use the GPIO pins and whether you need to put a HAT directly on top of your Pi or you are comfortable using

a ribbon cable to connect to the pins, because some cases allow only that. The CSI camera and DSI display ports may also be blocked by some cases.

Cooling solution

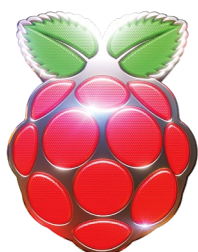
Some cases come with built-in fans, while others use passive cooling by turning the top of the case into a heatsink. Others leave room for a fan but don't come with one. Are you planning to use the case in the living room? Could fan noise become an issue? If so, a passively cooled Pi makes more sense. Cooling with the Pi 4 is more critical than for other models.

Special requirements

Some cases enable you to put the Pi on the back of a screen or come with a screen attached. Other models are meant to look or function like retro game consoles. The more advanced cases offer extra features, such as the Argon One M.2, which has connectivity for an M.2 SATA SSD.

The Pi 400

With the release of the Raspberry Pi 400, we see a modified Raspberry Pi 4 4GB inside a keyboard chassis. We have access to the GPIO and great cooling but no connections for the official Raspberry Pi Camera or touchscreen. If you are just starting out, the Raspberry Pi 400 may be the best all-round purchase.



» GET POWERED FROM AFAR

In the early days of the Raspberry Pi, keen makers just grabbed any old basic 5V plug-in USB power supply and off they trotted. But as Pi models started to get more and more power-hungry, such antics

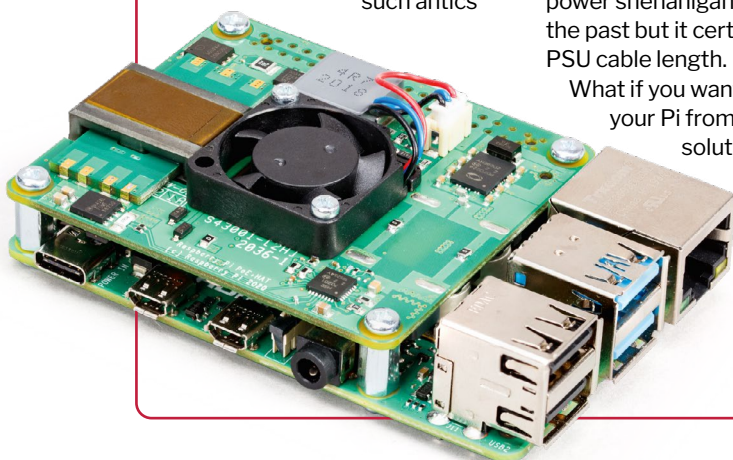
first brought funny colours and stutters to fully loaded Pis, then eventually blank screens. The latest Pi 4, with its USB-C power connector and official PSU supplying 15W, ensures such power shenanigans are a thing of the past but it certainly limits the PSU cable length.

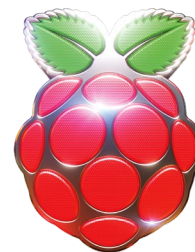
What if you want to power your Pi from afar? Battery solutions do exist but obviously are time-

limited. However, a cable solution is available in the form of the Raspberry Pi PoE+ HAT, standing for Power Over Ethernet, a standard for delivering device power from a compatible network router or switch over suitable Ethernet cabling. The official PoE+ HAT is designed for the Raspberry Pi 3B+ and Pi 4 with a maximum 20W, so there's spare for powering USB devices, too, while PoE Cat5 enables lengths up to 100m.

The official board isn't the only PoE HAT game in town; the WaveShare PoE USB Hub is a Pi Zero board that offers the same power possibilities alongside additional USB ports.

You can dangle your powered Pi from the end of 100m of Cat5, if you fancy.





Best all-round case

Argon Neo £15

The Argon Neo has attractive gunmetal-grey aluminium on its top and sides, with a shiny black plastic bottom. An aluminium plate covers the surface of the Raspberry Pi 4 itself and uses an included thermal pad to bond the CPU and RAM to the metal for passive cooling. There are cut-outs for the GPIO pins and the CSI/DSI ports. You can easily fit a standard-size HAT on the Argon Neo, without the need for an extender or a ribbon cable. There's even a little nook where you can prop up a Raspberry Pi Camera module.

Best project case

Pimoroni Pibow Coupe 4 £10

If your priority is ease of access to the GPIO pins so you can add a HAT, this could be your best choice. The Coupe is not completely sealed on top, so it leaves the GPIO pins unobstructed and provides enough room to run cables to both the camera and display CSI ports. If you like a dash of colour, the rainbow version will brighten up your desk, while different options for various Pi models are available.

Best smart case

SmartiPi Touch 2 £24

The SmartiPi Touch 2 enables you to combine the official 7-inch touchscreen (sold separately) with the Camera module and just about any Raspberry Pi board (except the Zero) to create an awesome kiosk-style device. You mount the official touchscreen in the SmartiPi Touch, which provides a space on the back to screw on your Pi and a camera hole as well.

Best advanced case

Argon One M.2 £40

A stylish but large piece of kit, which measures 140x93x40mm, the Argon One M.2 looks like a prop from *Star Trek* and is made with an aluminium top piece and a plastic lower section, where an M.2 SATA hard drive can be installed. The Pi 4 is placed sideways and custom boards break out the ports, control the cooling fan and provide a power button. A magnetic hatch on the top of the case provides access to the GPIO; next to this is a slot that is useful for routing the Pi Camera and official display cables through.

Best active cooling

Unistorm dual-fan case £15

This case both passively and actively cools the Pi 4 via a two-part aluminium construction and a pair of 20mm fans, which are powered directly from the 5V and GND pins of the GPIO. This does mean that HAT connectivity is blocked, but individual GPIO pins are available for use. The case offers access to ports, including making connections to the CSI and DSI before closing the case. Fan noise is low, and no louder than a typical laptop fan running at idle. **LXF**



Contributing for non-coders

Aaron Peters, who is a coding tinkerer at best, describes some ways that non-technical folks can contribute to open source projects.

One takes a village to run an open source project, or so the saying goes. Or would, if software were a thing back when that saying came to be. It nevertheless still applies to open source, which moves forward thanks to the efforts of contributors around the world.

Specifically, thousands of developers write millions of lines of code to create all the components that make up our favourite operating system. From the kernel to base utilities to desktops and applications, many of these contributors volunteer their time to improve and grow the FOSS ecosystem.

But even more people use the software, but aren't developers. If you, like the author, are among this group, how do you go about lending a hand? In this article, we'll explore a variety of ways you can pitch in to help an open source project, none of which requires

writing a single line of code. And what's more, the work you do can actually benefit you as well. It's a win-win!

Creating open source software requires a number of skill sets. We'll look at some of these in terms of the value they bring to the project, as well as what skills and tools you need to get started.

1 Quality assurance

Commercial entities have a fiduciary duty to their customers to produce software that's free of defects. Open source has a similar responsibility, but projects likely don't have funds to staff a quality assurance (QA) department with people dedicated to testing. Open source projects are always looking for contributors to help make sure the software is as bug-free as it can be.

An easy way to help is to report bugs you find in the software you use. Projects often manage their work in



KDE's Visual Design Group provides styles and imagery you can use to mock up your own idea.

issue-tracking systems. For example, the Fedora Project welcomes bug reports on the Red Hat Bugzilla site (<https://bugzilla.redhat.com>). If you find something wrong with the software you're using, you can submit a description of it to the dev team in a few simple steps.

Before logging your bug, make sure it isn't a known issue. It's a good idea to check any 'troubleshooting' or 'known issues' list that might come with the software you're using. You can search the Fedora list with the keywords that best describe your problem. Be as descriptive as you can, because the less precise the search, the more results you need to wade through. For example, if your system won't boot after an upgrade and you search simply for 'boot', Fedora's Bugzilla presents a list of over 1,500 items to slog through.

If, after some due diligence, you're satisfied you've discovered something new, you can take the following steps to submit your report:

1. Create an account in the Bugzilla system.
2. Fill out the form, making sure to select the correct component, version and other options. In the

» HOW TO START CONTRIBUTING

To make the most of your contributions, select a project you're passionate about. If you have a nine-to-five job, volunteering your free time can make life hectic. So, being truly excited about the project you're joining helps ensure your effort is time well spent.

To get started, poke around the project's website. Most have a page or section called Get Involved or How To Contribute featured prominently. Where exactly it exists differs, but as an example, the projects in this article welcome new contributors as follows:

FEDORA: The project has a site (<https://whatcanidoforfedora.org>) that uses simple questions to help determine the best way for you to pitch in.

LIBREOFFICE: The *LibreOffice* project uses the same format, available at <https://whatcanidoforlibreoffice.org>.

KDE: The Get Involved page on KDE's community wiki offers a more straightforward list of its teams (https://community.kde.org/Get_Involved), each of which has a page with its details.

MOZILLA: The Mozilla Foundation's Contribute page (www.mozilla.org/en-US/contribute) contains paths for you to find your ideal role in the community.

screenshot (page 46), note that 'Fedora' is the product, and 'emacs' is the component (a fun example, as we all know Emacs is perfect).

3. Fill out the Description as completely as you can. Simply saying "this is broken" is a good way to get your report tossed out.

4. Including screenshots where applicable makes it easy for developers to quickly understand the problem.

Once you submit the bug, you can expect a member of the project to triage it. This could involve asking you further questions, adjusting some of its settings (such as Severity if it's overall more or less important than your initial assessment), assigning it to a developer, or rejecting it. Note that a rejection isn't necessarily a judgement on whether it's valid or not. But it's common that the issue you're reporting is something that was introduced by the upstream project, who should rightly be the ones to perform the fix. You can then follow the progress of your reported issue from being assigned to fixed, and hopefully at some point released.

Reporting bugs is a simple but invaluable way you can help improve the quality of a project's software.

2 Design

Another way to assist free software projects is to contribute to their design. In this context, design has two separate but related connotations. The first is in the sense of software design – how an application should behave. Software projects in general do better when they listen to the requests of users, and open source projects are no exception. This is another great way to dip your toes into the volunteer waters, in that the only requirement is to give some real thought to your suggestion as you submit it. Even better, the process to do so is likely very similar to submitting a bug. The only difference may be making sure it's marked as an idea or feature request rather than a bug.

A more direct way to contribute to projects is UX/visual design – the way the software looks and how you interact with it. For example, in the KDE project, you're invited to join the Visual Design Group's (VDG) Matrix



chat room to learn about the team. Once you've talked with some of the other members and decided on the idea you'd like to submit, you flesh it out using two tools. Firstly, the VDG uses GitLab Issues to track in-process tasks. You can use this to describe the overall concept and hold a discussion about it using the built-in comments. In addition, you use a tool of your choosing to provide a mockup of your idea to help the team (including developers) visualise it.

The team offers samples and design assets in its Figma project to get you started. Finally, the VDG also welcomes help with creating icons and maintaining its Human Interface Guidelines, available on the KDE Developer wiki (<https://develop.kde.org/hig>).

3 Documentation

When software is developed well, who needs documentation, right? Wrong. Making sure software

has usable documentation can help prospective users evaluate it better, and gives the impression of a mature, complete product. And nearly every software project is happy to accept contributions to its documentation.

If you're interested in writing docs, each project's toolset involves a different level of complexity. For example, in addition to its formal application docs, KDE allows users to contribute to the UserBase wiki. This is as easy as registering for an account, deciding what to write about and what format to use (a how-to page or a full manual, for example), and contacting someone on the team to get started. While you're still expected to follow the project's workflow and style guide, you can pretty much start with just an up-to-date version of the component you're documenting and a web browser.

In contrast, other projects have a more technical infrastructure for their documentation. For example, the Ubuntu Documentation team follows a docs-as-code philosophy. In order to contribute, you need to take the following steps:

1. Install the tools, which include the DocBook markup language, XML transformation utilities (XSLT and FOP, for example), the Bazaar version control system, and Make to build everything:

```
$ sudo apt install bzip2 xsltproc libxml2-utils yelp-tools yelp-xsl make docbook docbook-xsl
```

2. Next, you need to register with and provide an SSH key to Launchpad, the management system used to coordinate Ubuntu and other open source projects.

3. Connect Bazaar to Launchpad with:

```
$ bzip2 whoami "Aaron Peters <address@example.com>"
$ bzip2 launchpad-login [username]
```

4. To contribute, you use Bazaar to check out a copy of the current doc set for the product you want to work on, one of either **ubuntu-docs** (for desktop Ubuntu), **xubuntu-docs** (for Xubuntu), or **serverguide** (for Ubuntu Server):

```
$ bzip2 branch lp:ubuntu-docs
```

5. Make your desired changes to the source files.

6. When you've finished, update to pull down any changes from other team members, then commit your changes locally:

```
$ bzip2 update
```

```
$ bzip2 commit
```

7. Finally push the commit back to Launchpad with a new branch name describing the changes you made:

```
$ bzip2 push lp:~[username]/[product branch]/[new branch name]
```

Please note that the above steps aren't intended to be a complete description of the process. You should follow the instructions on the Ubuntu Wiki's System Documentation page (<https://wiki.ubuntu.com/DocumentationTeam/SystemDocumentation>) if you decide to start writing for Ubuntu. Instead, they're intended to give you an idea of some of the technical steps you need to take. And it's important to look at these as a challenge, rather than a deterrent. One of the most rewarding things about contributing to open source is all the things you can learn along the way. Working on documentation is particularly beneficial, as you're teaching others while you learn yourself.

4 Translation

If you speak more than one language, translation is an extremely valuable way to give back to an open

Before reporting a bug, please read the [bug writing guidelines](#), please look at the list of [most frequently reported bugs](#), and please [search](#) for the bug. You may also use the [Guided bug entry page](#) for an easier step by step method.

Show Advanced Fields (*) = Required Field

Security: ☐ This report is security sensitive. Red Hat takes security very seriously and we aim to take immediate action to address any security related problems that involve our products or services. You should check this if:

- If you have found a security vulnerability with a Red Hat product or service
- If you are unsure about how a known vulnerability affects a Red Hat product or service

Alternatively, you may email the information to secalert@redhat.com. You can communicate with us securely using our GPG key.

* Product: Fedora Reporter: acplknd@gmail.com

* Component: emacs - Component Description

* Version: emacs-f Severity: unspecified

* Summary: emacs-s Hardware: Unspecified

emacs-vm OS: Unspecified

emacs-epc

emacs-epi

emacs-lua

emacspeak

Description: Comment Preview

Components can have different reporting requirements, therefore you must select a component before entering a description

Attachment: Add an attachment

Submitting a Fedora bug report is as simple as creating an account on Red Hat's Bugzilla and filling out the form.

» INTERACTING WITH THE COMMUNITY

Contributing to open source projects brings benefits over and above being able to see your work represented in your favourite bit of FOSS. One of the most rewarding aspects of volunteering is interacting with the community, including fellow contributors as well as users. In fact, starting out by just talking with other community members is a great way to get a feel for how the project operates, and whether it's the right fit for you. As you might expect, each project has its own preferred way of collaborating. But the following represent some places where you can start to interact with other volunteers, and maybe get some direction on how to get started.

MAILING LISTS: This decidedly old-school way is still very common among open source projects. Subscribe to the mailing list to receive threaded updates in your inbox.

IRC: Internet Relay Chat (IRC) is also still going strong as a communication method for many FOSS projects.

TELEGRAM: Some projects are moving to the Telegram protocol for their chat.

SLACK: Finally, some communities use Slack. While the limitations on its freemium service require some workarounds, its availability and familiarity make it an increasingly popular choice.

Once you've found the right channel(s), it's a great idea to simply lurk for a while. Many projects actually recommend you do just that. You'll get an idea of how the group operates, what opportunities are available and, importantly, where they may need some help.

source project. This is because with a comparatively small amount of work, you can open up a whole new market to the software. This isn't to say that translation is easy, but by converting a moderate amount of text in one language (typically English, for better or worse) into another, you can add the population of a whole country to the project's user base.

The Mozilla Foundation makes it extremely simple to start pitching in. A page on the Community Portal (<https://wiki.ubuntu.com/DocumentationTeam/SystemDocumentation>) provides links to start localising either software or help content. The process for both is similar:

1. Make an account in the appropriate system.
2. Gain an understanding of the project's guidelines.
3. Find things that need to be localised into your particular language.
4. Start submitting your translations.

If you choose to work on Mozilla software, for example, you'll use the project's *Pontoon* localisation platform. Most software designed for translating text works on the basis of strings, or individual fragments or sentences. *Pontoon* displays all the strings that require translation. Once you select one, there is a variety of tools at your disposal to help you create the translation.

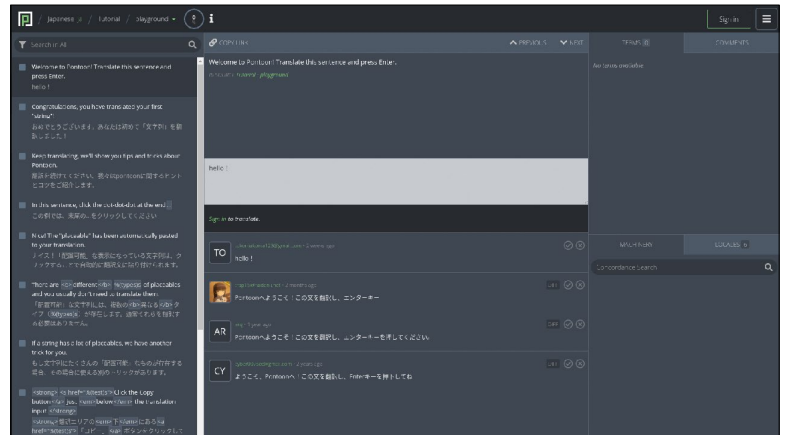
As shown in the screenshot (above-right), you start by selecting a string from the list on the left. This opens it at the top of the centre column. The grey area below is where you place your translation of that string. Any previously-submitted translations for it appear at the bottom of the centre column. The right-hand side contains tools and other hints for you to use while translating, including any terms with specific translations, available translations of identical or similar strings, and the results of machine translation. When you've finished, you can submit the translation, and then move on to another. Other volunteers subsequently review your translation for approval to be included in a project release.

5 Management and coordination

Even moderately sized open source projects can become complex, for a few reasons. Firstly, the project's membership is likely to be widely dispersed, right across the globe. This makes coordination challenging due to differences in time zone and, in some cases, language. Furthermore, unless volunteers are independently wealthy, they're juggling their contributions with some other occupation.

But over and above these logistical issues, the fact of the matter is that contributors are volunteering, and the process should be enjoyable. This makes the task of managing an open source project just as pivotal as it is for commercial projects, though for different reasons. If you're organised and can liaise between different groups, you can position a project so that others can focus on their core competencies (coding, for example) without having to worry about administrative matters. This enables these volunteers to be more efficient, and to focus on the contributions that make them happy.

We're fulfilling such a role in The Good Docs Project (<https://thegooddocsproject.dev>), an open source initiative to provide best-practice documentation guidance and assets such as templates. As the project



Mozilla's Pontoon localisation platform makes it easy to get translating quickly and provides plenty of tools to boot.

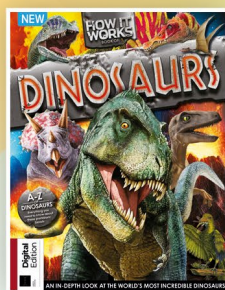
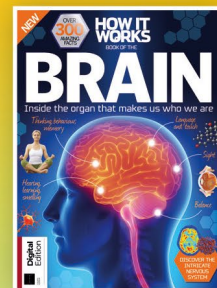
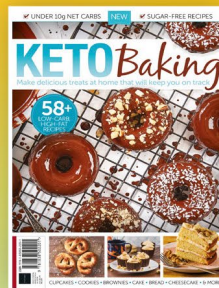
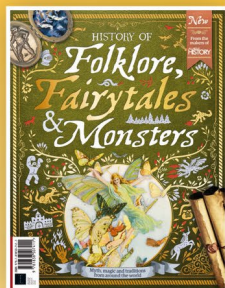
grew from roughly six to now several dozen active members, it quickly became difficult for the proverbial left hand to know what the right hand was doing. So we began to implement Agile product management principles alongside the writing of doc templates. As a result, we're now headed towards our second formal release (shown in the screenshot, below), and already have a backlog of items we can look forward to working on in the coming year.

Other functions, including advocacy, community relations and marketing, are also part of, or at least adjacent to, a more central coordination function. It's a great way to sharpen up your negotiation skills as well, because you don't have direct authority or monetary incentive as you might as part of corporate management. Instead, it's a chance to work with great people and use your organisational skills to make everyone else's volunteering a lot easier.

Hopefully, this article has not only laid out some of the options to contribute to an open source project if you're not a developer, but also provided an idea of what they take and what benefits they provide. Having volunteered in two of the above capacities (coordination and writing documentation), we can attest that it's an immensely rewarding experience that also gives you the chance to work side by side with great people. If you have the bandwidth, you should definitely consider giving it a try. It's an important part of what makes free software so great. **LXF**

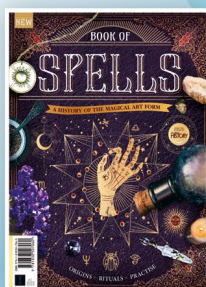
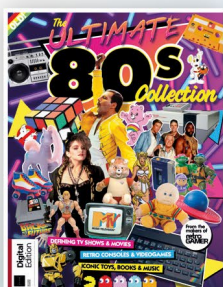


By rolling out management processes, The Good Docs Project now has a consolidated view of how releases are progressing.



DISCOVER OUR GREAT BOOKAZINES

From travel and history to gaming and photography, you're certain to find something you're passionate about

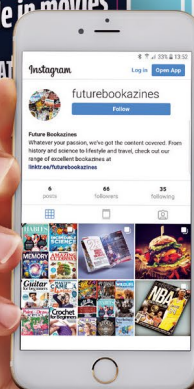


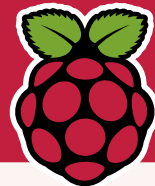
Follow us on Instagram  @futurebookazines



www.magazinesdirect.com

Magazines, back issues & bookazines.





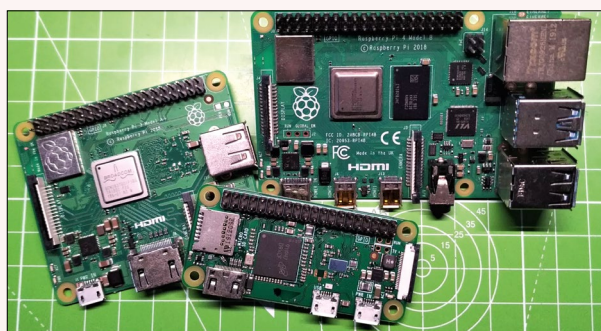
Santa Claus delivers extra Pis for Christmas

Saint Nick, aka Eben Upton, drops 100,000 extra Raspberry Pis on Tiny Tim just in time for Christmas.

Due to the delays of print publishing and extended Christmas deadlines, this news will seem a little late, but we're doing our best! Raspberry Pi CEO Eben Upton announced via the official blog that 100,000 units had been secured for single-unit sales (one unit per customer) and that by the second half of 2023, it is expected that there will be a return to pre-pandemic stock levels. This is good news for those who have felt the bite of the supply shortage that has dogged the Pi for well over a year. The bad news? Raspberry Pi Zero and Zero W see a \$5 price increase, but the Pi Zero will be available for bulk purchase in 2023.

Upton acknowledged the patience of the community and offered 100,000 units, made of

Raspberry Pi Zero W, 3A+ and Raspberry Pi 4 2GB and 4GB, for single-unit sale. He confirmed that units are "flowing into the Approved Reseller channel now, and this is already translating into better availability figures on rpilocator". You can read the full announcement at www.raspberrypi.com/news/supply-chain-update-its-good-news.



Hold off on weak imitations – the Pi should be returning as you read this.

Cool Pi 4

Not cool dude!

While the Pi shortage continues, plenty are stepping in to offer more expensive, less well-supported offerings. The latest is the dubiously named Cool Pi 4 – looking very similar to the classic Pi but with the eight-core RK3588 SoC and costing around \$140. Find out more from the Chinese site: <https://cool-pi.com/category/6/pi-4-model-b>.



CREDIT: Cool Pi

Aaeon Extreme!

Too much to handle?

Asus has a decent history of supplying SBC boards for development. Its latest is a beast, sporting a 12th-gen Intel Core i7/5/3 Celeron with up to 12 cores and 16 threads, PCIe V4, 2.5Gb Ethernet, up to 32GB DDR5, USB 4.0 and more. Find out more at www.aaeon.com.



CREDIT: ASUS



Neil Mohr
badgers his small children into doing electronic things so he can write about them. Is that so wrong?

» Pi-TERNATIVE LIFESTYLE?

The Raspberry Pi wasn't the first SBC on the scene. Long before, there was the very similar BeagleBoard, released in 2008. When the Pi was first released, many advocates pointed out that the BeagleBoard was better specified (it was) and had better support (it did at the time), though with its BeagleBone and Black edition, it was a good deal more expensive (\$89) than the \$25 Pi. BeagleBoard is still going strong and is currently gearing up to release a RISC-V-based BeagleV. It might not be an SBC as such but the Arduino Uno microcontroller just about predates the Pi and at the time had (and still does have) a huge open source community behind it, while being affordable and easy to access, though lacking the full-computer aspects of the Pi. Why am I telling you this? Pi models are currently hard to come by but that doesn't stop inventive companies offering similar products both in function and often very much in their looks. We're happy to cover that news, but should you be rushing out to grab one? It's something that I would avoid. I enjoy the immense amount of support the Raspberry Pi offers from its fabulous community – it makes it simple to troubleshoot and I'm not worrying about updates to crucial distros down the line. That's not to say at least some of these alternative boards don't have merit. BananaPi has stuck around and has gone from strength to strength, but is it helping to fund education research in the UK? I know which boards I'll keep buying, if for no other reason than that.

Petoi Bittle Robot Dog

The rise of quadruped robots is upon us — their mission to boost STEM and AI learning with **Anj Bryant**.

IN BRIEF

A modular beginner's robot kit that offers the ability for Scratch-level coding and enhancing the base kit with additional sensors, servos and more advanced coding languages.

Quadruped robots aren't new, but they're as cool and fascinating today as they were a few decades ago when they first surfaced. Bittle is a programmable robot dog targeted at robotic beginners ages 14 years and up, or anyone who wants to have fun learning and playing with robots. Priced at £310 (assembled), it comes programmed with a few starter tricks, such as walk and trot, that learners can try out to familiarise themselves with what the robot can do, then later expand to more complex types of behaviour.

This may seem like an expensive STEM kit, but it's fairly affordable compared to similar products. In fact, some servo-based quadruped robots, such as the Xgo Mini or the PuppyPi, retail for well over £500.

Bittle's open-source platform also enables you to add a Raspberry Pi and attach Grove sensors to extend its capabilities, explore AI machine-learning projects and try out various STEM experiments.

It ships in two models, a pre-assembled kit and a DIY construction kit, and three colour options: black and yellow, blue and yellow, or red and yellow. There are three ways to control your Bittle: using the IR remote that comes with the kit, downloading the mobile app on your phone, or installing the desktop app on your computer.

The pre-assembled robot also comes precalibrated and preloaded with a set of basic tricks such as walk, sit, stand and trot, which you can speed up or slow down using the controller of your choice.

It trots around naturally, is very agile and maintains good balance even as it changes direction. However, it isn't as graceful on carpet as it is on smooth surfaces.

Standing 11cm tall and weighing under 300g, Bittle is the perfect size to pick up and place in different settings. We found the head came off too easily, but this is a feature rather than a defect. The

head acts as a clip that can hold on to tiny objects and is also where the add-on sensors are placed.

Bittle's short battery life quickly became an issue. An hour isn't enough time for play and exploration at all. Extra batteries can be purchased for £32.

Construction kit

Bittle also comes in a do-it-yourself construction kit, which ships unassembled so you can experience putting the entire robot together yourself — a worthy and most enjoyable activity for anyone interested in robotics. You'll also learn circuitry and cable management.

The main body frame and legs are made of hard, durable plastic. It's cleverly designed with an interlocking mechanism, so parts easily snap in place. Assembly videos and tutorials are available online if you need help. There are also diagrams that you can refer to in the online manual. When you assemble Bittle yourself, you need to upload code to the NyBoard and make sure your connectors are connected to the right circuit.

To use Codecraft to program Bittle, make sure you have OpenCat 1.0, as version 2.0 is not yet compatible. You need to connect to your Bittle to check what version of OpenCat is installed. Use the USB adapter and dongle, Wi-Fi or Bluetooth to connect, and run the Petoi Desktop App. In the main window, click on Firmware Uploader.

You can download the Codecraft app on your computer or go to <https://ide.tinkergen.com> and select Bittle to start experimenting and programming different routines for Bittle. When done, click the Upload button on the left side menu. It takes a few seconds before Bittle executes your script. There are 16 online courses available for beginners and advanced learners. Bittle also supports Python and C programming languages.

The hands-on coding experience from the apps, free online coding courses and further expansion projects and experiments definitely make Bittle a worthwhile investment. Not to mention the support of the open-source online community. **LXF**

Sit! Stay! Now write an entire issue of *Linux Format*! Who's a good boy?



VERDICT

DEVELOPER: Petoi
WEB: www.petoi.com
PRICE: £310 (£299 kit)

FEATURES	8/10	EASE OF USE	8/10
PERFORMANCE	7/10	VALUE	8/10

A fun introduction to robotics and coding, but its short battery life, incompatible firmware and calibration issues could turn eager young learners away.

» **Rating 8/10**

Elegoo Neptune 3 Pro

Denise Bertacchi wishes everything in life could be relied upon to add more features while remaining excellent value.

SPECS

Build vol: 225x225x280mm
Material: PLA/PETG/TPU/ABS (up to 260°C)
Extruder: Direct drive
Nozzle type: 0.4mm MK8
Platform: PEI textured spring steel sheet, heated
Levelling: Automatic, inductive probe
Runout sensor: Yes
Comms: MicroSD card, USB Type A
Control: 4.3-inch colour touchscreen, removable
Size: 475x445x515mm, 8.1kg

Gone are the days of 'beginner' printers that welcome new makers with a trial by fire. The Neptune 3 Pro arrives 90% pre-assembled and goes from box to first print in 30 minutes.

Everything we liked from the Neptune 3 has been carried over and improved in the Pro edition. The PEI-coated build plate holds on to prints with the right amount of stick, then releases when cold. Bed levelling is a snap, and the Z height was on target without needing adjustment. The removable touchscreen is easy to use, and thumbnails have returned when you use Elegoo's slicing software. It also adds a few new goodies: a direct drive that really grips slippery filaments, such as TPU, dual Z screws for steady printing, and a built-in task light.

You get tools to build and maintain the printer, side cutters, a plastic scraper, two spare nozzles, and a microSD card with a USB adapter. There's also a small sample of white PLA to print your first model.

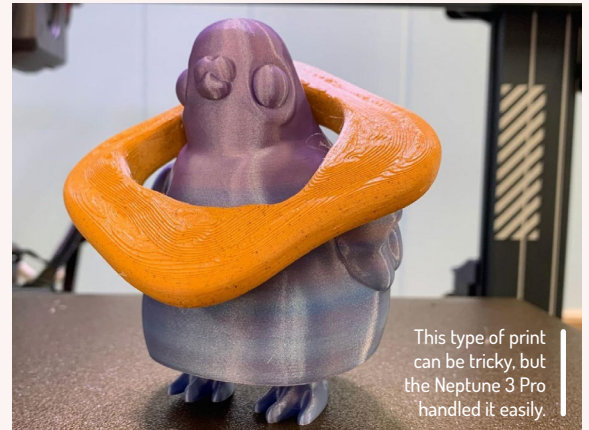
Hot off the press

The most interesting update is riding on the gantry: a dual-gear direct drive that does an excellent job of grabbing squirrely material like TPU. It has three cooling fans and an improved aluminium heatsink. Elegoo brags about the new "TC4 titanium alloy" throat pipe, but there's still a chunk of PTFE tube in there, so it's not an all-metal hotend. The improvements should help prevent clogs, but you're still limited to 260°C on the hotend – which is fine for most 3D printing enthusiasts anyway.

The power supply has been moved beneath the build plate to make room for a second Z axis screw. Shifting the weight lower and adding Z rods on both sides should help with print accuracy, especially on taller prints.

The Pro has a large removable touchpad with a magnetic base. The graphic menu is easy to understand, although the 'buttons' could be more sensitive.

Elegoo has also put some work into the auto bed levelling system, which now taps 36 points before



This type of print can be tricky, but the Neptune 3 Pro handled it easily.

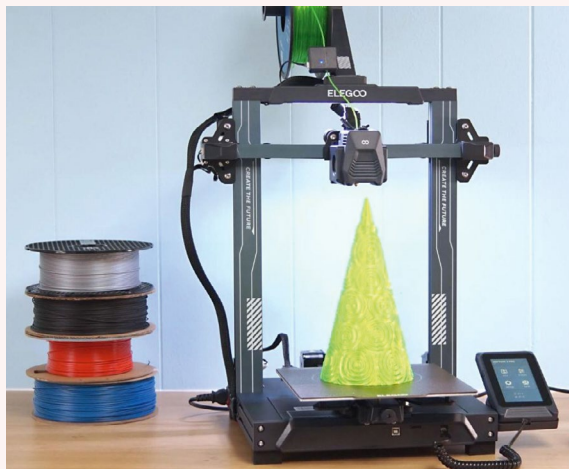
making its calculations. It uses a hard-mounted build surface and an inductive probe that works amazingly well.

The Neptune 3 Pro is a direct-drive printer with its extruder and hotend all in one spot. To load filament, tap Prepare on the main menu, then Temperature and select the one of four presets that match your filament.

It comes with a copy of *Elegoo Cura*, which has profiles for all its machines. It also has the added ability to create a thumbnail of your model to display on the printer. This function doesn't work if you use *OctoPrint*, as the Raspberry Pi overrides the main screen. Standard versions of *Cura* or *Prusa Slicer* don't have profiles for any of Elegoo's printers, but you can use the profile for an Ender 3 Pro and adjust the build volume accordingly.

The Neptune 3 Pro printed wonderfully right out of the box. The first print was the pre-sliced Buddha Elegoo provided on the microSD card. Using the supplied sample PLA it turned out very smooth.

The Neptune 3 Pro is a great 3D printer for people who want a polished machine that's easy to use straight away. Its simple auto bed levelling, PEI-coated flex tftplate, and excellent direct drive all help beginners by eliminating the major headaches we often experience with stripped-down basic printers. **LXF**



The Elegoo Neptune 3 Pro comes mostly built and ready to go.

VERDICT

DEVELOPER: Elegoo
WEB: www.elegoo.com
PRICE: £244

FEATURES	9/10	EASE OF USE	9/10
PERFORMANCE	9/10	VALUE	9/10

The easy-to-use Neptune 3 Pro offers upgraded features, like a direct drive and dual Z axis, without an upgraded price.

» **Rating 9/10**

Pi ROBOT

Pi robots: motors and control

Part Two!
Don't miss
next issue,
subscribe on
page 16!

In part two of his series, **Les Pounder** shows us how to safely connect and control motors with a Raspberry Pi.

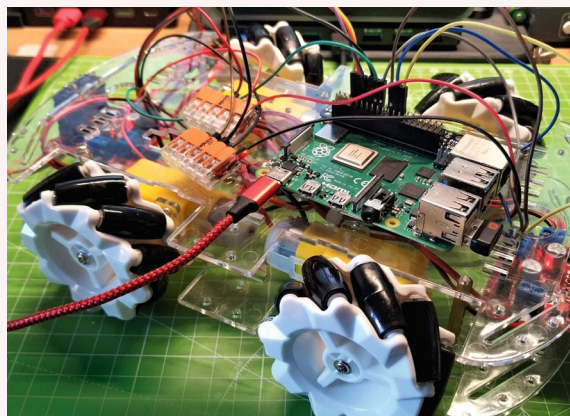


**OUR
EXPERT**

Les Pounder is associate editor at Tom's Hardware and a freelance maker. He blogs about hacks and makes at <http://bigLes.com>.

At the end of part one, we had built our robot from a kit and soldered wires from the motor terminals. In this part, we move on to writing code that will control the motors and move our robot. As our robot has four motors, we need to use two motor controllers. One for the rear motors, the other for the front. But which are the best choice for this project, and how do we interface the Raspberry Pi to them? We will create our own Python module (a library of prewritten code) to control the motors and create a test script to ensure it all works.

This part can get quite complex – we are working with four motors. Take your time, tackle just one motor at once, and double-check your connections before moving on. Sure, we want to get to the bit where our robot runs around under our command, but we need to put some work in before we reach those giddy delights.



It may look a mess, but this robot is ours. We're still in the debug phase, so making it look pretty is not a concern yet.

YOU NEED

- > Robot kit
- > Pi 3B+ or 4GB
- > 2x L298D motor controllers
- > 12x female-to-female jumper wires
- > USB battery for testing
- > Code: <https://github.com/lesp/Linux-Format-Robot/archive/refs/heads/main.zip>

Motor controllers

We can't just connect the GPIO of the Raspberry Pi to the motors. The amount of current consumed by a motor is many times what the Pi GPIO can produce. Instead we need something that sits between the Pi and the motors: a motor controller. The controller receives the signal from the Raspberry Pi's GPIO and triggers its output pins to send power to the motor. In reality, the motor controller is an H bridge, where changing the state of two pins can send the power to alternate motor pins. This forces the motor to spin in either direction, giving us directional control. The most common motor controllers for makers are the L9110S, L298N and DRV8833. But which one do we need?

To determine this we need to look at the datasheet for our yellow DC motors. It seems they have a max current of 800mA. This is the stall current, which is the current consumed by the motor when it is forced to stop – a wire may catch in the axle, or the robot becomes trapped. Looking at the datasheet for the L9110S, we see that it can handle 800mA, but only for one motor; two motors could see 1.6A go through and destroy the controller. We tested the stall current for one motor and noted that it reached 900mA, which is a definite strike against the L9110S.

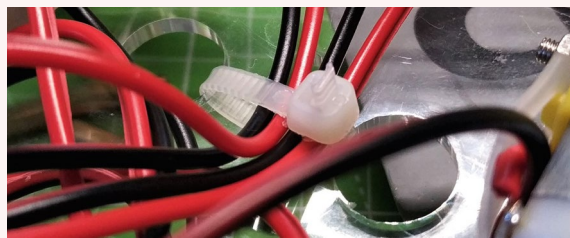
That leaves us with the DRV8833 and L298N. The DRV8833, a chip from Texas Instruments, has a

maximum current, per motor channel, of 1.5A. Two motors running from one unit could pull 1.8A in a worst case scenario. So for now, the DRV8833 is a maybe.

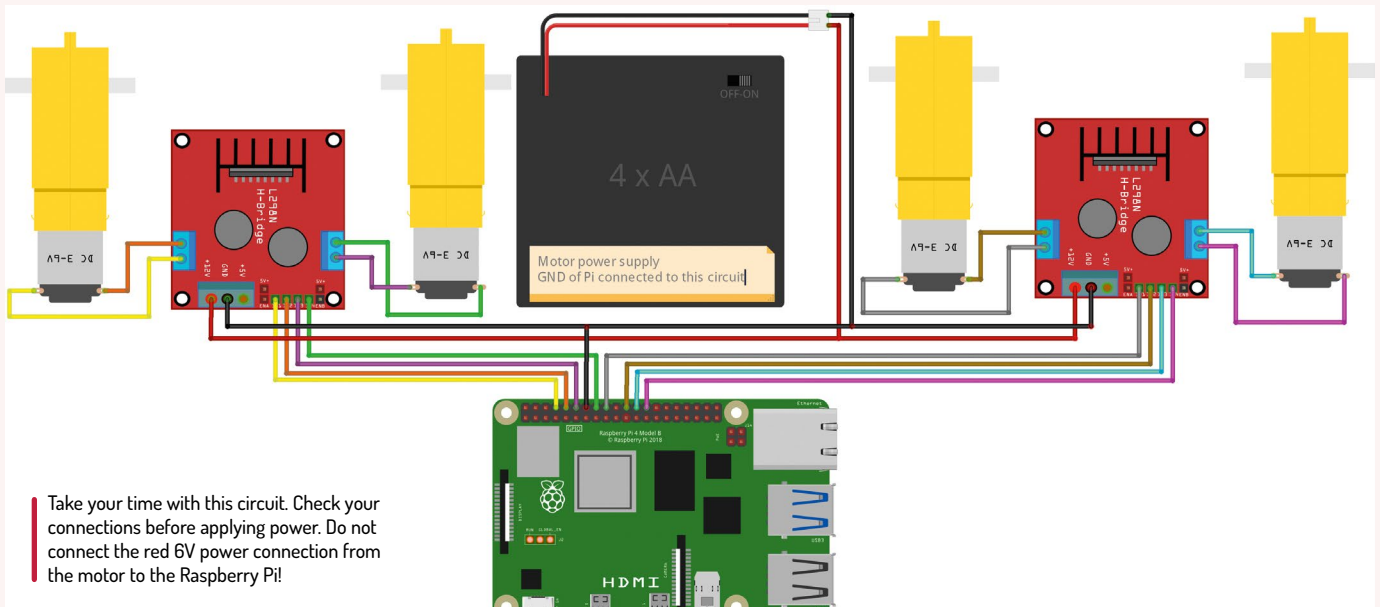
The final motor controller is the L298N, which is a beast of a motor controller. The L298N is the name of the chip itself, but it is commonly sold as the L298D, which features a breakout board for screw terminals and header pins. You'll also notice that the L298D has a massive heatsink. Well, it needs it as this chip can sink 2A per motor channel, over double the stall current of a single motor. This makes the L298D a great choice for this robot project. We are going to need two: one for the front motors, the other for the rear.

Connect the motors

Wiring up the motors to the controllers is an involved process, so let's do it one by one. We start with the rear



Cable ties are your new best friend. They can keep your components tight to the chassis, and clean up the mess of wires.



motors. To wire up one motor we need the two wires from the motor connected to OUT1 and OUT2 on the L298D. It doesn't matter much which goes in OUT1/OUT2; we'll test that later. From the Raspberry Pi, connect GPIO14 to the IN1 pin, and GPIO15 to IN2. These pins control the motor rotation. Now we have one motor connected to the L298D, and our Pi is also connected.

Connect the second motor to OUT3 and OUT4, and GPIO18 and GPIO23 for the IN3 and IN4 pins. Connect the power from the batteries (or USB battery) to the 12V and GND of the L298D via a breadboard that we'll use to break out one connection into many. We also need to connect any GND from the Pi to the GND of the motors; use another series of breadboard connections for this. The batteries only power the motors, not the Pi. Four AA batteries is 6V, much more than a Pi can safely use. Right now we have two L298D motor controllers, powered from batteries/USB, with their GND pins connected to the GND of the Pi and the power source. This gives us a common ground reference necessary for the circuit to operate correctly. These two motors will be the rear-left and rear-right of the robot.

Now for the front motors, and for this we need another L298D. This time, connect the front-left motor to OUT1 and OUT2, then connect GPIO24 and 25 to IN1 and IN2 respectively. Connect the front-right motor to OUT3 and OUT4, and then connect GPIO8 and 7 to IN3 and IN4, respectively. Again connect the 12V terminal to the batteries via the breadboard, and connect GND to the breadboard row of GND pins. An alternative to a breadboard are Wago 221. These are that turn one connection, for example GND, into many GND connections. We can use a Wago 221 with five connections to common the GND of our Raspberry Pi and the battery. Another Wago 221 can be used to break out the 5V connection from the battery to both L298D motor controllers. See the picture (page 55) for an example of how we did this.

Testing time

Before we can set our robot off on an adventure, we need to ensure that we have full control of the motors. To do that we create a test script that checks that each motor can spin, and that we have control of the direction and speed. This test script is based upon GPIO Zero, an excellent Python module from Ben Nuttall and Dave Jones. GPIO Zero simplifies working with the GPIO by abstracting the complexities into human-readable instructions.

Begin by powering up to the Raspberry Pi desktop. From the main menu, go to Programming and select *Thonny*. In a new, blank window, save the code as **robot-test.py**. We start by importing two modules. The first is GPIO Zero, specifically the **PWMOutputDevice** class from it. This is used for fine control of the motor

QUICK TIP

Wire up each motor individually. Check the connections match those in the table (see page 54). If your motor spins the wrong way, swap the connecting wires between the motor and the L298D.

» MOTOR CONTROL BOARDS

We've focused on bespoke H bridge motor controllers as the method to work with motors on the Raspberry Pi. They are not the only option, though. There are many other motor boards to choose from and here we highlight two that we have personally used. First is Pimoroni's Explorer HAT Pro, a board that was used in the Raspberry Pi Foundation's Picademy teacher training course. This versatile board has two motor outputs, capacitive touch inputs and 5V tolerant inputs. It is a great board for newcomers and while it isn't for use with big motors, for 6V micro gear metal motors, it is perfect.

The other controller is Cytron's Maker Drive, which is a breakout for an MX1508 motor controller. This tiny board is made for makers who want to get things to move. It has solid screw terminals, header pins for GPIO connections and, in a truly innovative move, push buttons. The four push buttons are used to test that a motor is connected and working. Press the button and it sends current along a set path, press another button and the H bridge flips and spins the motor in the opposite direction. We've used this board in a few projects, even with a Commodore 64 and a GPIO breakout to write Basic code to make a motor spin.

» **GET YOUR Pi FILLING HERE** Subscribe now at <http://bit.ly/LinuxFormat>

QUICK TIP

If you can't get any L298D motor controllers, a DRV8833 will do the job. The DRV8833 can cope with 1.5A of current per motor channel, but it is a little harder to work with.

speed by sending pulses of current instead of simple on/off. The second module is time, and we use the **sleep** function to create pauses in the code:

```
from gpiozero import PWMOutputDevice
from time import sleep
```

Our focus now turns to creating objects that act as an interface between the code and the GPIO pins that control the motor. The objects are given the cryptic names of **rra** and **rrb**. These refer to rear-right motor, and then one of two terminals represented by a and b. Object **rra** is the rear-right motor terminal a, connected to OUT1 on the L298D. This is controlled from GPIO14 on the Pi. The value of GPIO14 is set to 0 so that the motor doesn't automatically start. We repeat the process for **rrb**, connected to OUT2, IN2 and GPIO15.

```
rra = PWMOutputDevice(14,initial_value=0)
rrb = PWMOutputDevice(15,initial_value=0)
```

Here is the code for the left-rear motor, connected to OUT3 and OUT4 via IN3 and IN4 and GPIO18 and 23.

```
lra = PWMOutputDevice(18,initial_value=0) #left rear motor terminal a
lrb = PWMOutputDevice(23,initial_value=0) #left rear motor terminal b
```

Here is the code for the two front motors.

```
fra = PWMOutputDevice(24,initial_value=0) #front right motor terminal a
frb = PWMOutputDevice(25,initial_value=0) #front right motor terminal b
fla = PWMOutputDevice(8,initial_value=0) #front left motor terminal a
flb = PWMOutputDevice(7,initial_value=0) front left motor terminal b
```

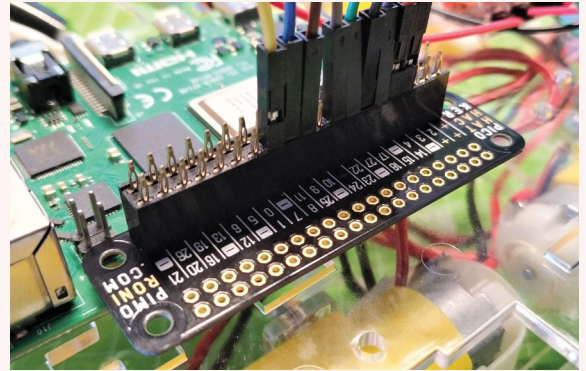
Save the code before moving on.

For quick reference, below is a table of connections between the motors, L298Ds and Raspberry Pi GPIO.

To control the motors, we create functions for forward, backward and spinning motion. Each function is named after the direction in which we want to go. The function takes an extra parameter, an argument, which is the speed at which we wish the motors to move, using a value between 0 and 1.

We start by moving forward using the **forward()** function. To move forward we need to spin the motors in the same direction. We set motor terminal a of each motor to run at the speed value we pass to the function. Motor terminal b is set to 0 so the H bridge in the L298D knows which direction the current flows.

```
def forward(speed):
    rra.value = speed
    rrb.value = 0
    lra.value = speed
```



We used a Pimoroni Pico HAT Hacker to help us remember the GPIO reference for each pin.

```
lrb.value = 0
fla.value = speed
flb.value = 0
fra.value = speed
frb.value = 0
```

Next we create a function called **stop()** to stop all the motors dead. No gentle deceleration – this is an emergency stop. It takes no argument.

```
def stop():
    rra.value = 0
    rrb.value = 0
    lra.value = 0
    lrb.value = 0
    fla.value = 0
    flb.value = 0
    fra.value = 0
    frb.value = 0
    print("ALL STOP")
```

Next we make a new line and write a few lines that call the functions to test that everything is working. We set **forward** to full speed, 1.0, but this can be any value above 0.5 for the test. Any values lower may cause the motor to stall, but later we can drop our speed down to lower than 0.5 while the motor is moving.

```
forward(1.0)
sleep(10)
stop()
```

Save the code and ensure the motor power supply is powered on and that the red LEDs on the L298D are lit. Lift the robot chassis up and ensure the wheels are clear of any objects and fingers. In *Thonny*, click Run and the motors spin. Check that each motor is spinning in the correct direction. If not, make a note and when the code stops, swap the motor terminal connection to the L298D for that motor and then try again.

Add more directions

With the successful test done, we add functions for backward, spin left and spin right. Place these below the **forward()** function, but above the test lines.

Backward

```
def backward(speed):
    rra.value = 0
    rrb.value = speed
    lra.value = 0
    lrb.value = speed
    fla.value = 0
    flb.value = speed
    fra.value = 0
```

Object name	Motor pin	Motor input	Pi GPIO
rra	OUT1	IN1	14
rrb	OUT2	IN2	15
lra	OUT3	IN3	18
lrb	OUT4	IN4	23
fra	OUT1 (2nd L298D)	IN1 (2nd L298D)	24
frb	OUT2 (2nd L298D)	IN2 (2nd L298D)	25
fla	OUT3 (2nd L298D)	IN3 (2nd L298D)	8
flb	OUT4 (2nd L298D)	IN4 (2nd L298D)	7

```
frb.value = speed
```

Spin left

```
def spin_left(speed):
    rra.value = speed
    rrb.value = 0
    lra.value = 0
    lrb.value = speed
    fla.value = 0
    flb.value = speed
    fra.value = speed
    frb.value = 0
```

Spin right

```
def spin_right(speed):
    rra.value = 0
    rrb.value = speed
    lra.value = speed
    lrb.value = 0
    fla.value = speed
    flb.value = 0
    fra.value = 0
    frb.value = speed
```

Now we can modify the test lines to use each function for five seconds:

```
forward(1.0)
sleep(5)
backward(1.0)
sleep(5)
spin_left(1.0)
sleep(5)
spin_right(1.0)
sleep(5)
stop()
```

Save the code, pick up the robot and click Run to test. Does every motor move correctly? If so you have completed the test and have a basic user-controllable robot. If not, swap the motor terminal connections between the motor and the L298D, and retest. In the



A Wago 221 connector can turn one connection into many. We used it to make a common ground reference and break out the 5V power.

download for this project is a quick reference table (Motor-Direction) of the values for each motor terminal.

Create a Python module

With a successful test under our belt we now move on to create a Python module that controls the robot. This is the easy bit. Copy all the lines from the imports and functions (we don't need the test lines), and paste them in a new *Thonny* window. Save it as **robot.py**.

Open a new *Thonny* window and use import the robot module and the **sleep** function from time:

```
import robot
from time import sleep
```

Add a few test lines that call the functions from the module and use them to control the robot. We preface commands with **robot** to tell Python which module to use. Save as **test.py** in the same directory as **robot.py**.

```
robot.forward(1.0)
sleep(5)
robot.backward(1.0)
sleep(5)
robot.spin_left(1.0)
sleep(5)
robot.spin_right(1.0)
sleep(5)
robot.stop()
```

Lift up the robot and click on Run. The robot should move in the same pattern as our previous test script.

To run the code remotely we need to start an SSH server. From the main Raspberry Pi OS menu, go to Preferences > Raspberry Pi Configuration. In the new window go to Interfaces and enable SSH. Hovering the mouse over the network icon in the top-right shows your IP address. On another computer, open a terminal window and use SSH to remotely connect to the Pi. You can use your IP address or hostname to connect. Typically the hostname is **raspberrypi.local** and the username is **pi**. So we would use SSH as follows.

```
ssh pi@raspberrypi.local
```

or

```
ssh pi@YOUR-IP-ADDRESS
```

Run **test.py** using Python (pick up the robot):

```
python3 test.py
```

And relax...

We now have a robot that we can code using a custom Python module. Next time we'll interface sensors into the robot to give it a level of intelligence that will see it navigate a room without outside help. **LXF**

QUICK TIP

Writing a test script is essential. We need to make sure that we have full control of the motors before we unleash the robot. Getting the basics correct will yield better results for future parts of this series.

» RASPBERRY Pi ALTERNATIVES

The Raspberry Pi is not the only brains for a robot. It just depends on what level of intelligence you need in your bot. At the most basic level, we can use a £2 microcontroller, such as an ATtiny85, to build a £10 Arduino robot: <https://bigl.es/friday-fun-attiny85-10-robot-challenge>. If we want to use MicroPython or CircuitPython, we can build a robot based around the Pi Pico or Pico W. We can even use a TensorFlow

Lite model to give the robot intelligence to identify objects and obstacles, and react accordingly. The Pi is the sweet spot for computational power and price, but if money is no object, Nvidia's Jetson range of boards has superior AI and computer vision capabilities. The Nvidia Jetson has additional cores devoted to machine learning, and with a webcam, we can train our robot to detect faces and objects, hopefully with altruistic

intentions. Another option is Khadas's range of boards. The VIM4 is a beast of a board, with 8GB RAM and an octa-core Arm CPU. It runs Ubuntu 22.04 and is priced just over the currently high prices of the Pi 4. The Khadas Edge 2 Pro comes with up to 16GB RAM and a SoC that features additional computing power for machine learning. Sure, we pay more, but we get much more power for robotics in a tiny form factor.

Tackling terminal tabular table tools!

With the right tool at his disposal, **Shashank Sharma** has no trouble making sense of vast heaps of tabular data.



OUR EXPERT

Shashank Sharma is a trial lawyer in New Delhi and an avid Arch user. He's always on the hunt for affordable geeky memorabilia.

Data from 2020 league results. You can use the mouse scroll wheel to navigate the file.

Vast amounts of information, separated into logical rows and columns that add up to a table, is the layman's definition of a data set. We see such data sets everywhere, from the ever-changing prices of consumer electronic goods to polling statistics, results of football matches, and more. With the right tool at your disposal, you can explore, clean, edit, summarise, analyse and plot the data into graphs for quick assimilation. While working with spreadsheets or CSV data is not traditionally envisioned as a task suited for the command line, *VisiData* will quickly dispel you of such notions.

Written in Python and released under the GPLv3, *VisiData* is incredibly fast and lightweight. You can easily use it to work with data sets featuring a few million rows of data. The project's goal for future releases is to be able to handle hundreds of millions of rows of data.

For our money, one of *VisiData*'s best features is that it's non-destructive. This means that it doesn't alter your raw data by default, leaving you free to manipulate the data to best analyse it without fear

of changing it in any fashion. In fact, it's also possible to save a *VisiData* session so that your particular manipulations are recorded, and you can then restore the session without having to start again from scratch. All this and more will become obvious as you go through this tutorial and, more importantly, spend some quality time working with *VisiData*.

Sit at the table

Some distributions, such as Arch Linux, carry *VisiData* in their software repositories. However, the quickest way to install the latest version is to use *Pip*, Python's package manager.

If you already have *Pip* installed on your distro, run the `pip3 install visidata` command. You can check the version of *VisiData* with the `vd --version` command. The project is constantly under development and this tutorial is based on version 2.10.2.

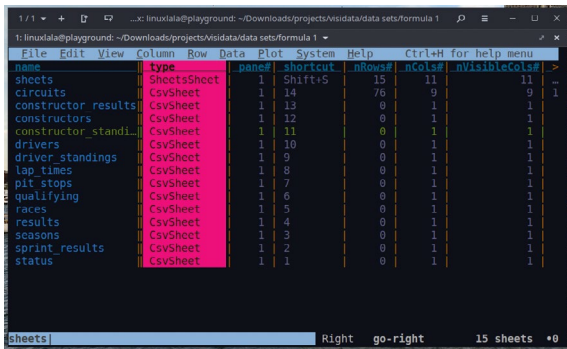
Ubuntu and Fedora users can install *Pip* using the software repositories by installing the **python3-pip** package with the `sudo apt install python3-pip` or `sudo dnf install python3-pip` respectively.

In addition to CSV files, *VisiData* supports a number of other file formats as well. Out of the box, it supports SQLite-databases, JSON files, Unix directories, ZIP files and more. Head to www.visidata.org/docs/formats/ for a list of all the supported formats.

You have to install additional packages if you wish to work with *Excel* spreadsheets, *Postgres*, *MySQL* databases, XML or YAML files, and various others. For instance, the **xlr**d and **openpyxl** packages are required for working with *Excel* spreadsheets (XLSX files). You can install these with the `pip3 install xlr openpyxl` command. You'll similarly have to run the `pip3 install mysqlclient` command before you can connect *VisiData* with a *MySQL* database.

Like traditional spreadsheet applications, *VisiData*'s workflow also focuses on sheets. Every file you open in *VisiData* is treated as a sheet. If you open a single file with the `vd filename` command, you are essentially working with a single sheet. However, if you were to run

Round	Date	Team 1	Score	Team 2
4	Sun Oct 4 2020	Aston Villa FC	7-2	Liverpool FC
2	Sat Sep 19 2020	Everton FC	5-2	West Bromwich Albion FC
1	Sat Sep 12 2020	Liverpool FC	4-3	Leeds United FC
2	Sat Sep 19 2020	Leeds United FC	4-3	Fulham FC
2	Sun Sep 20 2020	Leicester City FC	4-2	Burnley FC
4	Sat Oct 3 2020	Everton FC	4-2	Brighton & Hove Albion FC
8	Sat Nov 7 2020	Crystal Palace FC	4-1	Leeds United FC
8	Sat Nov 7 2020	Chelsea FC	4-1	Sheffield United FC
3	Sun Sep 27 2020	West Ham United FC	4-0	Wolverhampton Wanderers FC
4	Sat Oct 3 2020	Chelsea FC	4-0	Crystal Palace FC
7	Sun Nov 1 2020	Aston Villa FC	3-4	Southampton FC
3	Sat Sep 26 2020	West Bromwich Albion FC	3-3	Chelsea FC
5	Sat Oct 17 2020	Chelsea FC	3-3	Southampton FC
5	Sun Oct 18 2020	Tottenham Hotspur FC	3-3	West Ham United FC
3	Mon Sep 28 2020	Liverpool FC	3-1	Arsenal FC
4	Sat Oct 3 2020	Newcastle United FC	3-1	Burnley FC
2	Sun Sep 20 2020	Southampton FC	2-5	Tottenham Hotspur FC
3	Sun Sep 27 2020	Manchester City FC	2-5	Leicester City FC
3	Sat Sep 26 2020	Brighton & Hove Albion FC	2-3	Manchester United FC
5	Sat Oct 17 2020	Everton FC	2-2	Liverpool FC



Shift+O opens the Options sheet, where you can tweak parameters such as the sidebar colour, the colour of the current column, and so on.

the `vd filename.csv filename1.csv filename2.csv` command, *VisiData* opens each file as a separate sheet. It only displays the first specified file by default. You can confirm this by looking at the bottom-right of the interface, which shows the name of the file that is currently open.

You can access all the other sheets by pressing **Shift+s** from anywhere in the *VisiData* interface. This opens a Sheets sheet, which provides a list of all open sheets. Use the arrow keys to select any other sheet and press Enter to open it.

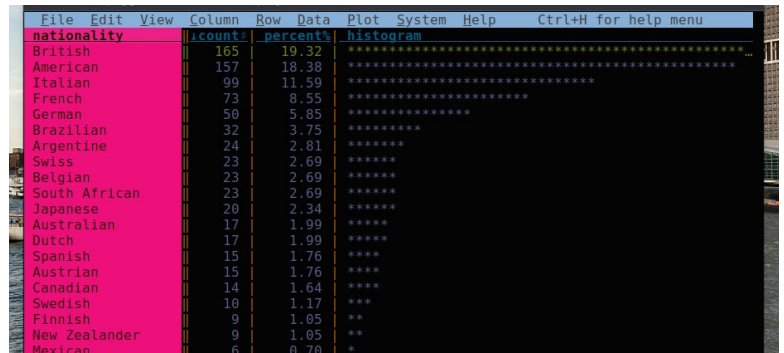
If you have a table open in *VisiData*, hit the **Shift+C** key combo. This opens a Columns sheet. All the columns in your table are listed as rows in the Columns sheet. You can configure different elements, such as the width and height of the columns, here. This is useful when your table has multiple columns where the data doesn't fit all on the screen, or if you wish to reposition columns in your table. Press **q** when you're ready to return to the source sheet.

You can similarly press **z** followed by **Ctrl+H** to open the Commands sheet. This displays all the commands available for you to use in the current sheet.

The Sheets sheet, Commands sheet and Columns sheet are examples of metasheets in *VisiData*. You can think of these as sheets that are designed to help you navigate and customise *VisiData*.

All the data files you open in *VisiData* are source sheets. Any manipulation that you perform on a source sheet in *VisiData* doesn't alter the original files.

Derived sheets are sheets that draw their data from source sheets, such as when you filter certain rows from a file based on a defined parameter, or create a



Using the frequency table, we can see that the UK has produced the maximum number of Formula 1 drivers, at 19.32%.

frequency table. We'll discuss both of these further down in the tutorial.

Working with VisiData

With *VisiData* installed, you can open any supported file with the `vd filename.csv` command.

Although *VisiData* is designed to be operated entirely with the keyboard, you can use the mouse for a few operations. When you open a file with *VisiData*, you'll see a menu bar at the top with typical entries such as File, Edit, View, Column, Row, Data, Help and so on. You can click on these to navigate submenus. You can also use keyboard shortcuts to access submenus. **Alt+h** for instance opens the Help menu, while you can access the Plot menu with **Alt+p**, and so on.

When you first open a file, the top-left entry is selected. You can navigate the rows and columns using the arrow keys. Alternatively, you can use **h** and **j** to move left and down respectively, while **k** and **l** can similarly be used to move up and right.

Keybinding	Function
h	Move left
j	Move down
k	Move up
l	Move right
PageDown/Ctrl+F	Move one page down
PageUp/Ctrl+B	Move one page up
gh	Move to leftmost column

QUICK TIP

VisiData's vast capabilities are all controlled using myriad keyboard shortcuts, which admittedly take some getting used to. But not because they are illogical or difficult to master. As a beginner, we recommend keeping a cheat sheet handy: <https://jsvine.github.io/visidata-cheat-sheet>.

» MAKING SENSE OF THE DATA

The quickest means to make sense of your data is by using *VisiData's* frequency table. Navigate to any column and press **Shift+F**. *VisiData* immediately offers details such as count and percentage, as well as a histogram to quickly assimilate the data.

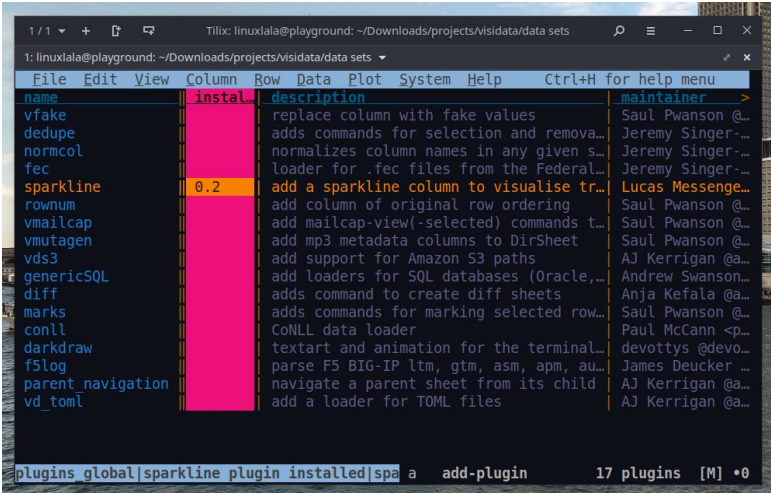
For some operations, such as when you want a frequency table as a combination of two different columns, you must make use of key columns. To set a key column, navigate to the column in question and press **!**. You'll now notice that the column is pinned to the extreme left of the sheet, irrespective of its original position. You can define multiple key

columns in a single sheet. To create a frequency table of all set key columns, press **gF**. Note that you have to press **g** followed by **Shift+F** to get this command to work.

A frequency table essentially counts the number of times an entry occurs in the given sheet. However, you can use additional aggregators such as sum, min, max, mean, median and others.

Navigate to a column and press **+** to add an aggregator. *VisiData* asks you to define the aggregator or press **Ctrl+X** to see a list. Note that you have to press **Ctrl+Shift+X** to access this list.





You can access the plugins installer by pressing Space and running the “open-plugins” command. Navigate to the plugin you wish to activate and press “a”. Pressing “d” deactivates a plugin.

gj	Move to last row
gk	Move to first row
gl	Move to rightmost column
/+<regex>	Search forward in current column
g/+<regex>	Search forward in all columns
?+<regex>	Search backward in current column
g?+<regex>	Search backward in all columns

As with the movement keys, **g** can be combined with many different *VisiData* keybindings. It denotes that the action is to be applied to the file as a whole.

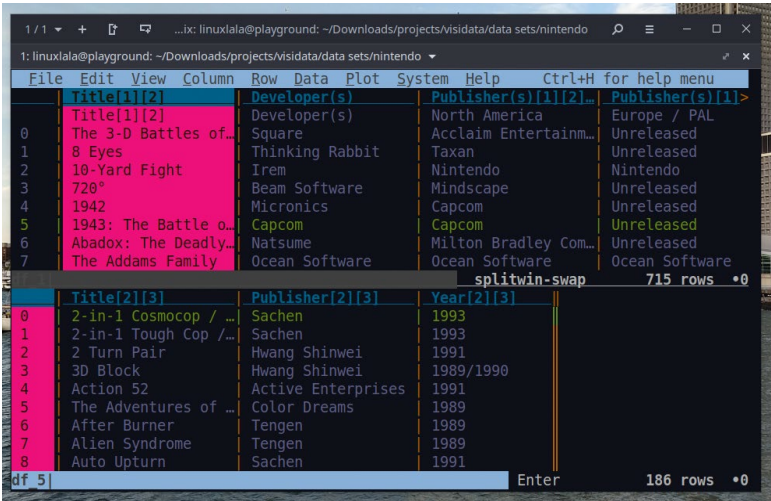
You can also search for data in your file using regular expressions (regex), which can be applied to a single column or all the columns in the file. *VisiData* drops you to the first row where it finds a match to your entered string. Press **n** to move the next matching entry, or **N** to move backwards.

Sort is one of the quickest operations you can perform. Select the column to sort and press **[** to sort the data in ascending order or **]** for descending.

We’ve already discussed how you can search for data in *VisiData*, but it’s also possible to select rows of data. In addition to manually selecting multiple rows, you can also select rows using regular expressions. Once selected, you can perform different operations

QUICK TIP

The split-screen feature, accessed by pressing Shift+z is ideal for accessing a sheet and its column sheet at the same time.



When working with multiple sheets at the same time, you can split the interface by pressing Shift+z to make life easier.

such as deleting all selected rows, or create a new sheet with only matching rows in it.

The following table lists some of the commonly used commands for selecting rows:

Keybinding	Function
s	Select the current row
u	Unselect the current row
t	Toggle selection
gd	Delete selected rows
<regex>	Select rows matching <regex>
\<regex>	Unselect rows matching <regex>

You can use **g** in combination with other keys for global effect. For instance, while pressing **s** selects the current row, you can select all rows by pressing **gs**. Similarly, **gu** unselects all rows, while **gt** toggles all rows between selected/unselected.

It’s also possible to move rows up and down a table, by pressing the **Shift+j** or **Shift+k** keys respectively.

You may also need to occasionally edit the cells in your file, such as when changing the *VisiData* settings. To edit a cell, navigate to it using the arrow keys or **h**, **j**, **k** and **l** keys if you prefer, and then press **e**. To cancel editing, press **Ctrl+c**.

When you exit *VisiData*, by pressing **q** – or **qq** if you have multiple sheets open at the same time – all the changes you made to the file, such as repositioning rows and columns, or editing cells in the table are lost. This is because *VisiData* is non-destructive by default and doesn’t alter your source files. You can choose to save an edited sheet as a new file or overwrite an existing file. Press **Ctrl+s** to save your current sheet. You’re prompted to provide a filename. The choice of extension, such as CSV or JSON, defines the format. Hit Enter when you’re done and the file is saved.

Making changes

Depending on your source files, you may on occasion encounter cells in your table that have content that doesn’t fit in the defined row and column size. When this happens, you can either scroll the contents in the specific cell, or alter the width of the columns and height of the rows.

You can alter the width of columns from the Column sheet. When you have a file open in *VisiData*, press **Shift+c** to open the Columns sheet. This displays all the columns in the table and their attributes, such as width, height, type, and so on. Navigate to the relevant entry under the Width column. Press Enter and then edit the value from the default 20 to another number that you think would suffice for the content in your file.

If you find the process of opening the Column sheet and changing the width of certain columns too tedious, there’s a quicker solution. Navigate to the column with the scrollable content and press **z** – this expands the current column’s width to accommodate the content. If you have several columns with scrollable content, you can press **g_** instead. As we’ve seen from various commands in this tutorial, using **z** with a keyboard shortcut means that it’s applied narrowly, usually to the current cell, row or column, while using **g** with a

» ESSENTIAL READING

VisiData is an incredibly robust and featureful utility and we can't cover a thorough usage in this tutorial due to space constraints. While many of its features, such as creating pivot tables, will be of use only to seasoned operators, this project features a lot of documentation that will be of use to novice and experienced users alike.

Your *VisiData* journey must begin with the 'An Introduction to *VisiData*' tutorial. This is split into different sections, such as The Big Picture, Basic Usage, Intermediate Usage and so on. Once you have *VisiData* installed on your machine, make sure you go through the Basic Usage section of the tutorial. This provides a good introduction to opening and closing files, saving sheets, manipulating rows and columns, sorting and filtering

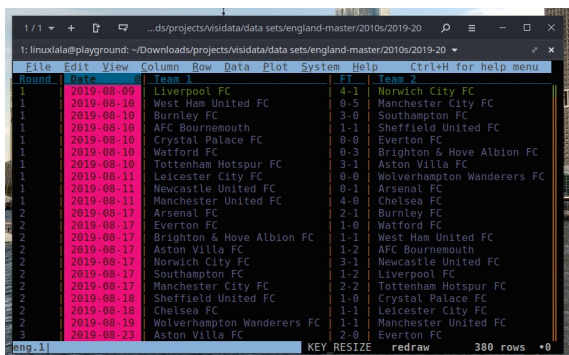
data, and more. The section on Intermediate Usage covers some of the advanced features, such as joining sheets and reshaping data by creating pivot tables, and more, while the Advanced Usage section is still under development. Even so, you'll appreciate the section on configuring *VisiData*. It's also possible to extend *VisiData* with plugins.

Although there's a dedicated subreddit for *VisiData*, you'll be better served raising queries on the GitHub page. Make sure you go through all the solved and open issues, as there's a good chance that your specific query has already been addressed.

For more detailed information, head over to www.visidata.org/docs to master the different components of *VisiData* and how to get the most out of them.

QUICK TIP

You can configure *VisiData*'s appearance using the `~/.visidatarc` file. This isn't created by default when you install *VisiData*. Head over to the project's GitHub page, and fetch a sample `.visidatarc` file from the Issues discussion page.



We've used the "g_" key combo to expand the width of columns and defined the date column with the @ keyboard shortcut.

keyboard shortcut affects the file globally. Of course if all you want to do is read the contents of a cell, without changing the width of the column, you can navigate to the cell in question, press `e` to enter the edit mode and use the arrow keys to move left and right to read through the contents. Press Enter when you're done, or `Ctrl+c` to cancel edit mode and return to the sheet.

Certain operations, such as drawing a graph or performing operations such as sum, median and so on, require numerical data in your columns. By default, *VisiData* treats all data in all columns as plain text. However, a typical file contains a variety of different data types, such as text, integer, date, currency and so on. You can define the data type for a column using the following keyboard shortcuts:

Keybinding	Function	Example data
#	Define the column type as an integer	1,12, 2005, 500000
@	Define the column type as date	09/10/2022, 2022-10-09, October 9, 2022
~	Define the column type as text	This is text

\$	Define the column type as currency	\$10.99, £25, €40
%	Define the column type as float	0.1, 29.99, -15.9

Depending on your choice, the corresponding symbol is displayed next to the column name. If the content of a cell can't be converted to the defined column type, *VisiData* shows a ! in the relevant cell.

We've only scratched the surface of all that's possible with *VisiData* and how it can help you quickly assimilate and manipulate data files. You have to spend quite a bit of time with it, and with the documentation, before you begin to fully appreciate its capabilities.

That said, the non-destructive default behaviour means you can press `q` at any time to exit the tool when you feel lost, without fear of affecting the source data. More than anything else, the freedom to explore without constantly worrying about destroying your files is essential for mastering any project with a steep learning curve. Refer to the Essential Reading box (above) to see where you can get started. **LXF**

You can use the edit mode (`e`) to change the name of a sheet from the Sheets sheet or the name of columns from the Columns sheet.

File	Edit	View	Column	Row	Data	Plot	System	Help
name	type	pane#	shortcut	nR				
sheets	SheetsSheet	1	Shift+S					
circuits	CsvSheet	1	14					
constructor_results	CsvSheet	1	13					
constructors	CsvSheet	1	12					
constructor_standi...	CsvSheet	1	11					
drivers	CsvSheet	1	10					
driver_standings	CsvSheet	1	9					
lap_times	CsvSheet	1	8					
pit_stops	CsvSheet	1	7					
qualifying	CsvSheet	1	6					
races	CsvSheet	1	5					
results	CsvSheet	1	4					
seasons	CsvSheet	1	3					
sprint_results	CsvSheet	1	2					
status	CsvSheet	1	1					

» ENHANCE YOUR TERMINAL-FU Subscribe now at <http://bit.ly/LinuxFormat>

Plan and write your next smash-hit novel

Nick Peers reveals how to develop better stories with the help of our favourite freemium app for budding novelists.



OUR EXPERT

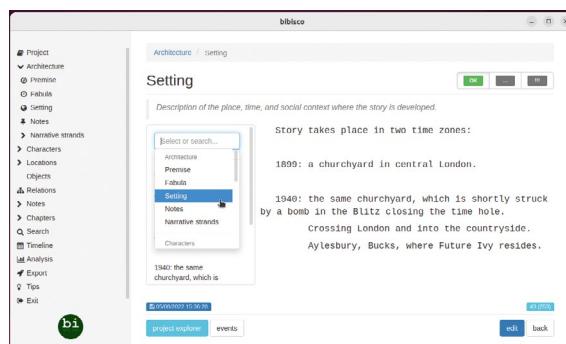
Nick Peers has one particular story idea knocking around his head – *Bibisco* could be the tool to finally bring it to fruition.

Anyone who's written – or attempted to write – a novel will appreciate just how difficult it is without doing your research. And where do you put all that research so it's easy to access when bringing it all together? Linux is blessed with all manner of apps promising to bring order to your story-writing chaos – we rounded up five in **LXF294**, and settled on *Bibisco* as the best of a very promising bunch. So, let's find out how it can help you with your next writing project, whether fiction or non-fiction.

Package me not

Sadly, *Bibisco* doesn't exist in any repository – you need to head to <https://bibisco.com> and scroll down to the Download section, where you'll find a choice of two downloads: the free Community Edition or the paid-for Supporters Edition, which costs a minimum of £24 (including VAT). All the core functionality can be found in the Community Edition, which includes a 30-day free trial of the Supporters Edition, so we'll clearly signpost any paid-for features we think are worth exploring.

Assuming you're starting with the free Community Edition, after clicking the Get It! button you're prompted to supply your email address. Once entered, click Get and you'll see a Show Content button, behind which you can find the downloads you need. Before doing this, you're prompted to create an account – this



Bibisco's Project Explorer tool enables you to view any or all of your notes, research and other info while editing.

is recommended as it will give you easy access to updated versions in the future.

Once done, you'll see a list of downloads (*Bibisco* is also available for Mac OS and Windows), and here you have a choice of three: DEB, RPM and ZIP. Ubuntu users will, of course, find the DEB package the easiest to work with – once downloaded, open the file in Software Install and follow the prompts.

First-time setup

Once installed, open *Bibisco* via the applications launcher. Read through and accept the licence terms,

» BIBISCO'S TEXT EDITOR

When planning and writing your novel, you have ample opportunity to get acquainted with *Bibisco's* rich text editor. It's not as powerful as a word processor, but contains all the key elements you need. Styling is limited to bold, italics, underline and strike-through, plus a highlighter, but you can define a default display font, size, lining spacing and so on via the settings button. There are also handy undo/redo buttons.

You'll also see a full-screen button, which switches *Bibisco* to distraction-

free mode, but note this is an SE-restricted feature. When in full-screen mode, there are no formatting controls on view, so make use of the usual keyboard shortcuts, such as Ctrl+B for bold text, and press Esc to revert back to the normal windowed view.

As you start typing, you'll see *Bibisco* is set to automatically save your changes as you go, but there's also a manual save button, too. You'll see a handy word count in the bottom-right of the screen, and once you've finished

typing your premise, you can change its status using the three buttons in the top-right – !!! (to-do), ... (not complete) and OK (completed) – to track progress.

You'll also see a Project Explorer button beneath the text editor window. Clicking this enables you to display the contents of any other section – from architecture to characters and chapter summaries – alongside the editor window, providing an easy way to refer back to your notes as you type. It's accessible from any text editor window.

then select your language – English (UK) – to begin. You're now prompted to select a folder for your projects (typically a subfolder inside **Documents**) and a place to store backups.

After clicking Finish, *Bibisco* quickly sets up the user dictionary, then prompts you to create your first project – you'll also see options for importing an existing project and accessing the main settings (it's worth taking a quick tour to set your theme and zoom level). Other settings include changing language and user directories, and tweaking *Bibisco*'s backup options.

Let's work our way through the new project wizard: click Create The First Project, where you're asked to give your project a name and enter its author – you can also change the project's language from here. Click Save, acknowledge the warning that once the language is set, it's set for the life of that project, and click OK.

Project goals

You find yourself at *Bibisco*'s main screen – as the annotation reveals, it's primarily a two-paned affair, with a navigation pane on the left. The Project screen in time lists a complete summary of your project, but for now, it simply lists your novel's name, plus 'This is your novel' in large letters and two buttons beneath.

The Here Are Some Tips button is a shortcut to the Tips section, where you'll find lots of writing advice, from developing believable characters to tapping into *Bibisco*'s planning tools to help develop your novel's background to improve your story's quality and realism.

The other button – Goals – is your first experience of accessing a feature restricted to the Supporters Edition (SE). As you're in trial mode, a warning pops up revealing how many days are left in your trial, with buttons to purchase the SE or continue (after a five-second countdown) to the restricted feature.

Clicking Continue reveals the Goals screen, where you can set a deadline for finishing your project (click the calendar to select year, month and day in turn), plus an overall word count. You can then set your own arbitrary words-per-day goal or click Calculate to have *Bibisco* do it for you. Click Save if you decide to make use of this, or Back to return to the previous page.

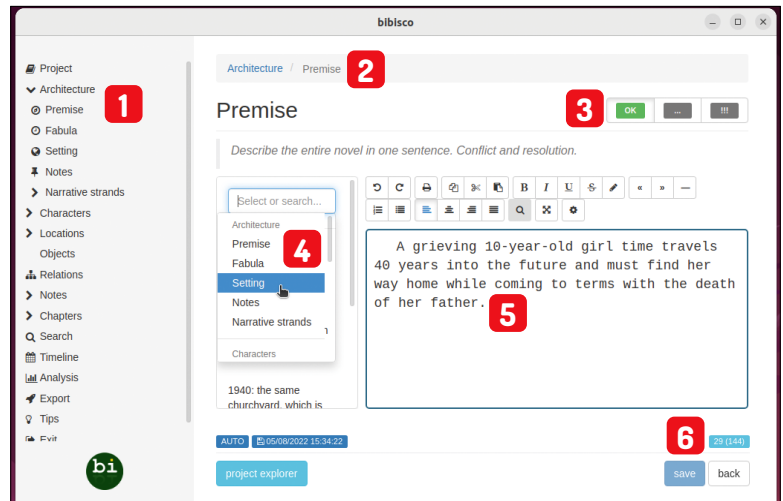
What's your premise?

Bibisco stores key information about your novel in its Architecture section, where you'll find four elements called Premise, Fabula, Setting and Notes, plus a Narrative Strands section beneath. If you click > next to Architecture, you'll see these options all appear in a submenu beneath Architecture, making them easier to access. Of these, only Notes is restricted to the SE.

Each section is accompanied by a short description and an icon indicating its current status (the red !!! icon indicates it's still in the to-do phase. Click anywhere inside the Premise box and you jump to its own dedicated section, where you're tasked with describing your novel in a single sentence. Click Get Started again to reveal a rich text editor, which works as you'd expect (see the box, left, for details).

The other key elements of the Architecture section – Fabula and Setting – work in a similar way. Setting provides you with somewhere to record a more detailed summary of your novel, while Fabula describes itself as a 'narration of the events stored by their logical

GET FAMILIAR WITH BIBISCO



1 Navigation pane
Use this to quickly access all of *Bibisco*'s tools – look out for collapsible > headings.

2 Breadcrumbs
The breadcrumb navigation system makes it easy to jump back up the currently selected section.

3 Status buttons
Use these to help track the progress of individual parts of your project, from planning to entire chapters.

4 Project Explorer
Click Project Explorer to toggle this box on and off, letting you access notes easily.

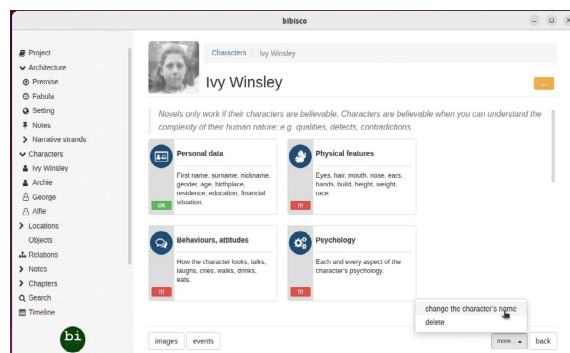
5 Rich text editor
Bibisco's text editor contains all the core functionality you need – click the Settings button to change display font and formatting.

6 Word count
Bibisco provides running word (and character) counts of each separate part of your project.

and chronological sequence', which translates as the place to record a more detailed breakdown of your story – you might be eyeing up the Timeline section at this point, but aside from the fact it's an SE-only feature, it's more an analytical tool, as you'll see later.

Beneath the four basic architecture sections are narrative strands. Click Create Narrative Strand to start your first one. This pops up a prompt to give your strand a title, which creates a standalone element that works in a similar way to the main architectural element: click it to open the text window for composing or editing. Narrative strands are linked to resolving so-called conflicts, which basically represent the characters' motivations (both internal and external).

As with all parts of *Bibisco*, you can set these up now or come back to them later. Navigating between the different elements is nice and simple, thanks to the expandable navigation pane on the left, but clicking



Build up a rich portfolio of information and history about each character to create fully rounded personalities.

QUICK TIP

Because *Bibisco* is distributed as a DEB or RPM package, you need to update it manually: either click the link in the original email you received or log into the Gumroad account you optionally set up during registration to access the latest version.



QUICK TIP

It bills itself as a novel-writing app, but Bibisco can be adapted to other purposes. It's great for family historians looking to write a memoir of an ancestor, or you could use it to write a general history, with real people filling the role of characters in your 'story'.

Back steps you back to the last thing you were doing, while a breadcrumbs system at the top makes it easy to move up the current section's navigation tree.

Have character

Every story has a protagonist, and *Bibisco* enables you to record as little or as much detail as you need about the people in your novel via its Characters section. Here you'll see characters are broadly split into two types: main and secondary.

Start with the star of your novel: click Create The First Main Character. Give them a name and click Save. As with strands, this creates a blank character template, ready for you to fill. You can add other character templates now or click your main protagonist to start turning them into a flesh-and-blood person.

A wide range of traits can be recorded: personal data covers the basics, while you'll find sections on physical features, behaviours and attitudes, psychology, ideas and passions, sociology and even a

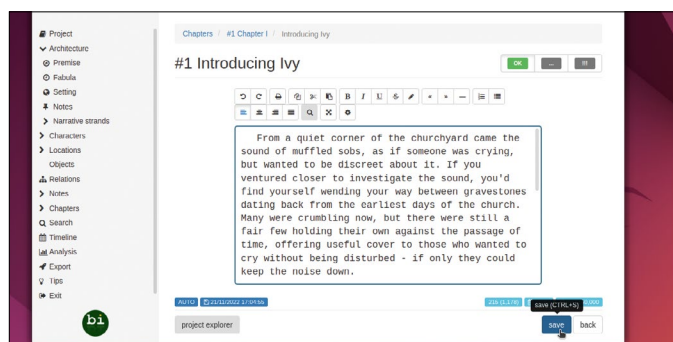
section on personal history dubbed Life Before The Story's Beginning.

Unlike the architecture section, many of these provide you with a series of questions to answer – click Edit and you'll find yourself in the familiar text editor, albeit with one key difference: a drop-down menu at the top enables you to move between each question. Obviously, all sections are optional, and you'll see an option at the bottom enabling you to switch between interview and free text if you'd rather write a biography. Again, use the status buttons above the text editor to help track your progress.

Things are simpler with secondary characters – you're given a free-form text editor to record all the key details you need about that person on a single page.

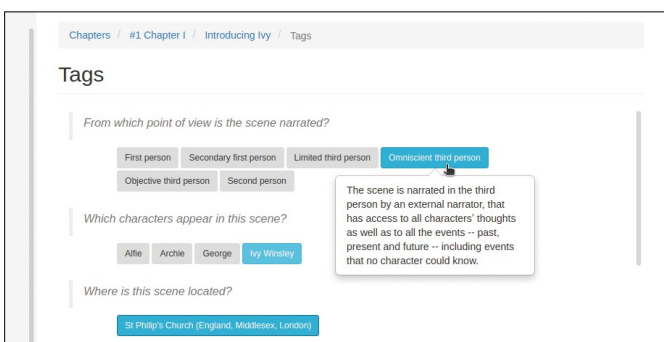
As with other sections, the Supporters Edition offers bonus features – you can attach images to characters via the images button (or + button next to their thumbnail for a profile picture), plus record date and time-stamped events, which show up in the

WORKING WITH SCENES



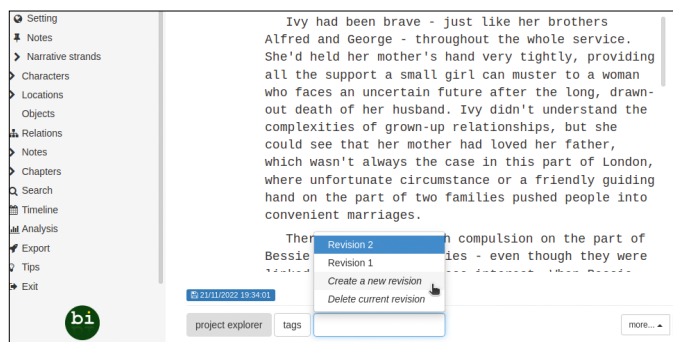
1 Set up a scene

Click Create Scene within the chapter you wish to place it in. Give your scene a suitable title and click Save. It appears under Scenes with the usual embellishments: title, word count and status icon. Click the scene to open its main page, followed by Get Started to start typing out the prose that will make up your novel. When finished, click Save followed by Back



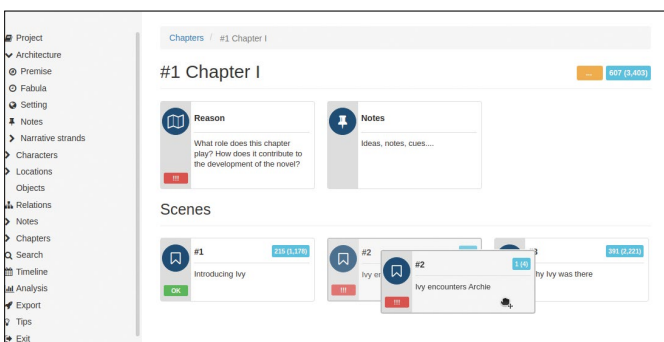
2 Add tags

If you want to make the most of your previous research, click the Tags button on a scene's main page to label each scene and inform better analytics. The first lets you record who's narrating the scene, whether first, second or third-person. Other tags link to your characters, locations, objects and narrative strands, plus you can assign a time and date to the scene.



3 Make multiple revisions

Not sure how a scene will progress from its current point? Look for the Revision drop-down menu on the main Scene page – click this and choose Create A New Revision. Click OK to use your current revision as a starting point for this version or Cancel to start from a blank canvas. You can now develop alternative storylines, and switch between (and delete) revisions as required.



4 Rearrange scenes

Another benefit of working with multiple scenes is that you can move them around. A simple reordering of scenes within chapters is done through drag and drop on the main Chapter page – just click, drag and drop a scene into its new desired place. Moving a scene to another chapter is done via the More button on the Scene's own page but is a Supporters Edition-only feature.

Timeline view, providing you with another means of structuring your story.

Record more research

The Locations tab is where you record details of the various places where your story takes place. Available fields are Nation, State, City and Name (effectively a description), plus a free-form page of text. SE users can also attach images and events, as with characters.

There's also an Objects tab for key items in your story, with the same level of detail as Locations – this entire category is restricted to the Supporters Edition, so while you can access it using your free trial, it's a minefield of pop-up nags. You also need the SE for the Relations section, where you can draw relationships between characters in diagram form, and Notes, an area to store miscellaneous ideas, cues and so on.

You're probably champing at the bit to stop planning and start writing your story at this point but, you'll see shortly the benefits of taking the time to record as much preparatory detail as you can. However, now is the time to let your creative juices flow.

Writing your story

Your actual novel is stored in just one part of *Bibisco*: Chapters. Things start by clicking Create Chapter to set up your first chapter – give it a suitable title (Chapter I if you're feeling uninspired, but each chapter is numbered for easy reference), and then click the chapter to access its own page.

Here you'll find separate sections covering Reason (for recording the chapter's role in your overall story arc) and Notes (unlike the other instances of Notes in *Bibisco*, this one is not locked behind the SE paywall). And beneath this is the chapter's contents, which is split into scenes. You can write an entire chapter within a single scene or subdivide it into multiple scenes. Follow the walkthrough to discover how to make the most of the feature – in particular take a close look at tags, which are essential should you wish to take advantage of *Bibisco*'s analysis tools (see the *Track Your Progress* box).

As always, *Bibisco* also offers some additional goodies to the Supporters Edition – the ability to add prologues and epilogues to the start and end of your novel, plus SE users can add an additional layer of organisation above the chapter level in the guise of parts. All three options are accessed via the More... menu under Chapters. Once you've created your various parts, you're prompted to choose which part to store each new chapter when giving it a title.

Going forward, you can reorganise your chapters by dragging and dropping them within the main Chapters screen – if you've set up parts, you can drag them between these, too.

Publish and be damned

There's no way to preview your novel as a single, continuous narrative within *Bibisco* itself; instead, you need to export it to another format, which is the process you'll follow once your manuscript is complete and you're ready to share it with others or want to

» TRACK YOUR STORY'S PROGRESS

As your story unfolds, you'll sometimes want to stop and take stock of where you are. Switch back to the Project tab for some handy statistics and progress bars to help you see how far you've come (and how far there still is to go). Purchase the Supporters Edition and the goals section stands out, as it provides you with an overall progress bar as well as a guide to how well you're keeping up with any self-imposed deadlines or word counts.

Make use of tags in scenes and you unlock all kinds of useful information via the Analysis tab. You get a breakdown of chapter lengths, plus detailed stats about the contribution from each character, location, object and narrative strand. Available stats cover overall distribution across the novel of individual character and object appearances, with date and time stamps, making it easy to ensure characters and objects don't appear in two places at once.

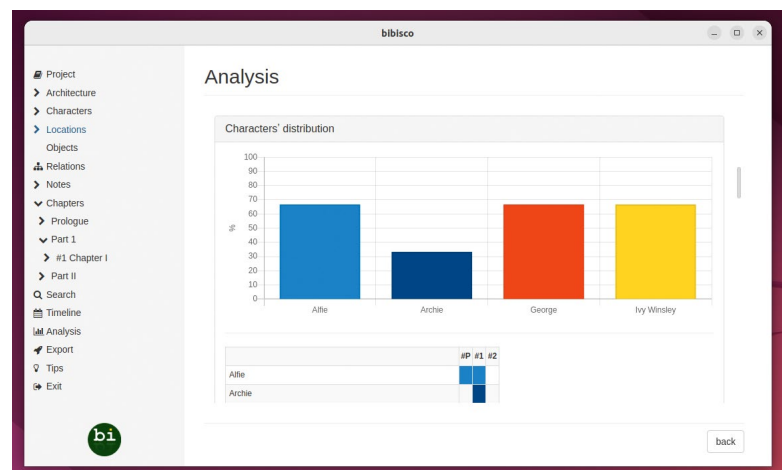
Supporters Edition users also gain access to the Timeline, which provides an alternative – and more detailed – view of your story's chronology. Whereas the Analysis section is limited to character and location appearances, you can also track the progress of all chapters, scenes and narrative strands from a single timeline. This enables you to plot your story's progress down to the exact minute, helping you to properly pace your novel (and ensure there are no glaring time-related plot holes).

produce a copy to print. Unsurprisingly, this involves a trip to *Bibisco*'s Export page via the left-hand navigation pane.

Here you'll find options for exporting your work in PDF, *Microsoft Word*, plain text and a ZIP archive (an additional EPUB option is – you guessed it – only available in the Supporters Edition). The ZIP archive is for sharing with other instances of *Bibisco*, so you'll probably want to choose PDF or *Word* (which can, of course, be opened in *LibreOffice Writer*).

After selecting your chosen format, you can apply various settings, from providing your name as the author to formatting the work, with such options as chapter title format and position, scenes separator, plus some optional extras in terms of formatting if required. You're also given the option of exporting the entire work or just a subset of it. Finally, choose your export directory and let *Bibisco* do the rest. **LXF**

Take the time to tag your scenes and it helps you to unlock a raft of useful analytic data.



» **GET A LOAD MORE FICTION...** Subscribe now at <http://bit.ly/LinuxFormat>

Polish your video footage with Kdenlive

Michael Reed looks at how you can use Kdenlive to produce high-quality videos suitable for uploading all over the internet.



OUR EXPERT

Michael Reed hasn't quite broken into Hollywood yet, but at least he can make a movie using Linux and open source software.

Kdenlive is an open source application that gives you everything you need to edit video footage into finished compositions. To this end, you can add video files from various sources, alter their length and position, and add effects to give the finished production a professional look.

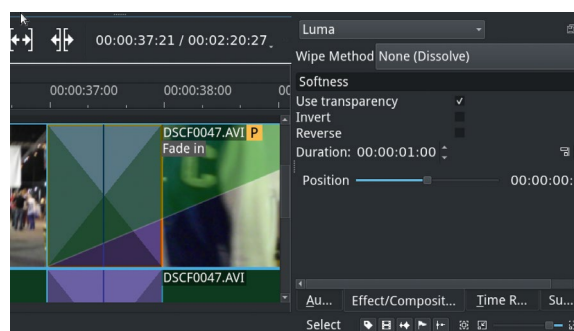
It's what is known as a non-destructive editor, which means that you don't alter any of the underlying files. Instead, data pointers within the program are updated, so you have almost unlimited scope to alter things and make fine adjustments at any stage of the process.

What we're imagining is the sort of project where you have a video of an event or a trip. You'll want to combine multiple video files and edit them down while adding some music, voice narration and title cards.

Lights, cameras, edit!

We assume you have the latest version of *Kdenlive* installed and you have some recorded footage you would like to edit. The first task is to start a new *Kdenlive* project (File > New), and this launches the project configuration dialogue. Select Custom Project Folder. Although video files are huge, it's a good idea to place them on your fastest drive, preferably an SSD. You can always move the files to a different drive, such as a network drive, after you have finished the project.

You can mix and match footage types within the project, but a *Kdenlive* project can only output a video



Here we're creating a transition between two clips on the timeline. On the right, we can edit its parameters.

file with a single frame rate and resolution. This means that it's impossible for *Kdenlive* to maintain consistent quality between clips that were recorded with different specs. For this reason, it's a good idea to choose specs for the project that match the majority of the footage that you are going to use. A shortcut here is to accept the defaults and drag and drop the most important video file into the Project Bin panel. This causes the program to ask you if you'd like to change the project settings to match the clip.

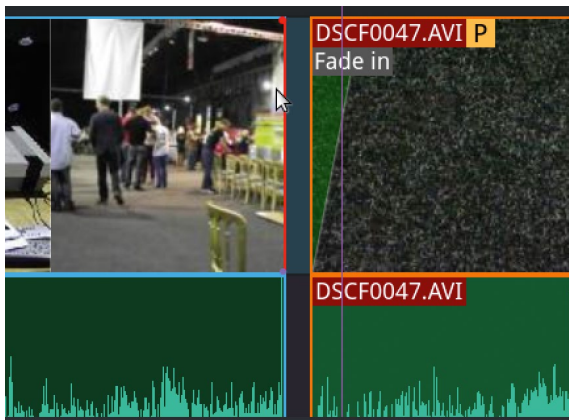
Because video files can be huge and require a lot of processing to handle, we recommend using proxy clips as they make *Kdenlive*'s editor far more responsive. Proxy clips are versions of the video clips that are stored at a lower quality and with less aggressive compression settings. As they are exactly aligned with the original video file, and because they are only used in editing, there is no difference in quality when the final output video file is produced. You can enable proxy clips in the Proxy tab of the Project Settings tab.

Once you have a project set up, it's time to start adding clips to the project. Typically, these are in the form of video files, image files and audio files, such as background music.

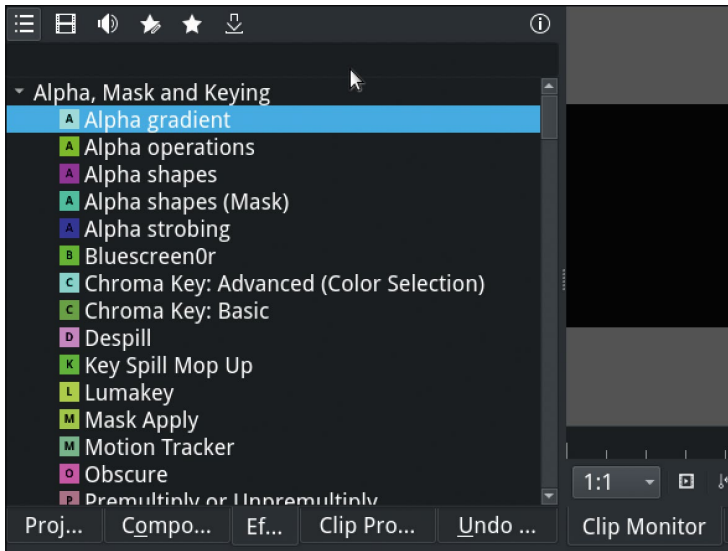
Drag the clips from a file manager into the Project Bin panel. If you've enabled proxy clips, as we recommended, the transcoding process begins in the background. At the same time, if there are files in a format that *Kdenlive* can't work with, you're asked for permission to transcode those as well. There's nothing to stop you from using *Kdenlive* at this point, but you're

QUICK TIP

Kdenlive is a well-known Linux program, and this means that it tends to be in the repository of most distributions. However, it's usually worth checking the version in the repository against the latest version on the website (<https://kdenlive.org>). AppImage, Flatpak and Ubuntu PPA are available



To resize a clip, hover your mouse pointer over the end of the clip, then click and drag.



As well as being categorised, the effects list is searchable. There's a tiny toolbar at the top to filter effect types.

better off letting it finish the transcoding operations for consistent performance.

We're on a Timeline

The basic workflow for creating a video in *Kdenlive* is to drag clips on to the tracks of the Timeline, where you can move them around and resize and cut them. In addition, multiple clips can be overlaid on adjacent video tracks. In situations such as this, tracks are evaluated in order, with the highest-priority video lane at the top. You might like to add a title clip with a transparent background on to a higher video track in order to introduce a new section of the video. Audio tracks, such as background music or narration, play simultaneously. Adjust the relative volume of such tracks while they are playing by selecting the Audio Mixer tab in the bottom-right side panel and adjusting the volume faders.

Click at the top of the Timeline to move the Timeline Cursor. This is the point from which playback begins if you press the play icon underneath the Project Monitor window. You can also start and stop playback by pressing Space. The Timeline Cursor is also used for other functions within *Kdenlive*. For example, if you move it to a position and press G, it places a useful guide marker that can be named and colour-coded – invaluable when you're navigating around a complicated project.

The mouse wheel scrolls the Timeline backwards and forwards, but note that this doesn't move the Timeline Cursor at all. This is handy when you're just checking things out, but it's annoying if you forget, press Space, and wonder why everything jumps back to the position of the cursor when you start playback. You can zoom all the way from a detailed view to an overview by zooming in and out (Ctrl+mouse wheel). The tracks are also collapsible, and it's worth collapsing tracks for more space on the Timeline unless you need the detailed view.

Clip to the Timeline

Kdenlive works with two main track types: video and audio tracks. Audio tracks should be pretty self-

explanatory, but video tracks include anything that can be seen, which covers video clips, still images and title clips. By default, *Kdenlive* provides you with four tracks – two audio and two video – but you can easily add more if required (simply right-click on the Master area of the Timeline and click Insert Track).

In recent versions of *Kdenlive*, if you drag a video clip from the Project Bin panel, the program automatically splits that clip into a video clip and an audio clip. Although they are separate clips on the Timeline from that point onwards, they are locked together initially, meaning that if you move one, the other moves with it. This is because they are grouped clips, but it's perfectly possible to ungroup them (Ctrl+Shift+G or Ungroup Clips in the right click menu) to cut, resize or delete

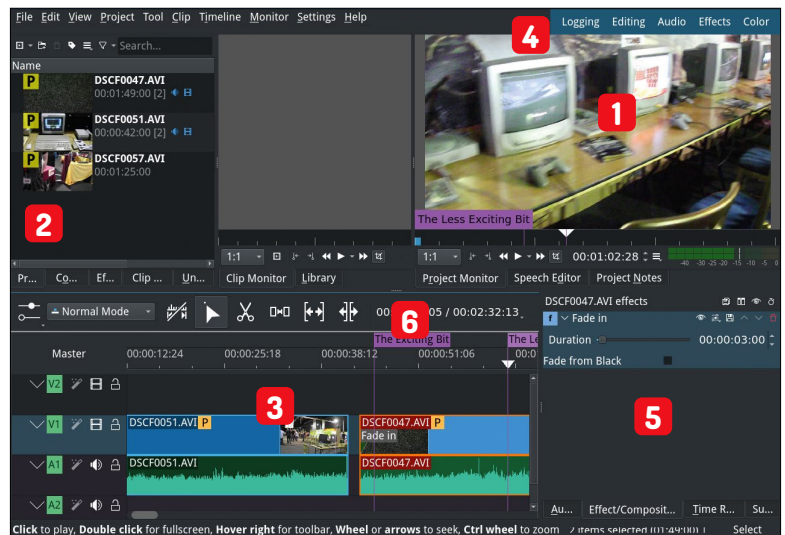
them separately. This is useful, for example, if you want to cut the audio picked up by the camera before the end of the clip.

Because *Kdenlive* is a non-destructive editor, nothing you do in this regard is irreversible. You could ungroup the aforementioned audio and video clips again and stretch out the audio part of the clip that you shrank before. The converse is also true; it is often handy to multiple-select clips (Shift+click) and group them (Ctrl+G) to make sure that they always move

QUICK TIP

Don't be afraid to drag the various areas of the user interface around. For instance, we found the initial size of the project monitor to be a bit conservative. Hover over the gaps between the different areas of the interface and do some pushing and pulling to experiment.

GET TO KNOW KDENLIVE



1 Project Monitor

This enables you to play back the current project. Like most *Kdenlive* panels, it's tabbed for multiple functions.

2 Project Bin

This is where you drop media clips that you would like to use in the current project.

3 Timeline

You can place clips, edit them and move them around on the Timeline.

4 Quick Layout Selector

The interface is helpfully tailored to suit the particular task in hand via the Quick Layout Selector.

5 Effect Editor

This is where you change parameters for an effect or transition.

6 Timeline Toolbar

One of many toolbars scattered throughout *Kdenlive*.



QUICK TIP

There are sites offering music that can be used freely in your videos. **Musopen** (<https://musopen.org>) has an emphasis on older recordings, such as classical music, while **Free Music Archive** (<https://free-music-archive.org>) offers original music released under the Creative Commons licence.

together, and it's usually worth regrouping clips after you've split them to make edits.

Moving and cutting

Returning to the Timeline Toolbar, by default the Select tool is activated. This enables you to select a clip with a single mouse click and move clips around by clicking and dragging.

One problem with an editor like *Kdenlive* is that it's impossible for clips to overlap each other on the same track, meaning that it can be awkward to move a clip on the Timeline if there are clips on either side. This is where the Spacer tool (Timeline Toolbar) comes in. Select it and try moving a clip that's surrounded by other clips, and you'll notice it shunts earlier or later clips to maintain their order. As your project becomes more complex, this is the main method used to move clips around. The Spacer tool even shifts clips above and below the selected clip to preserve the timings.

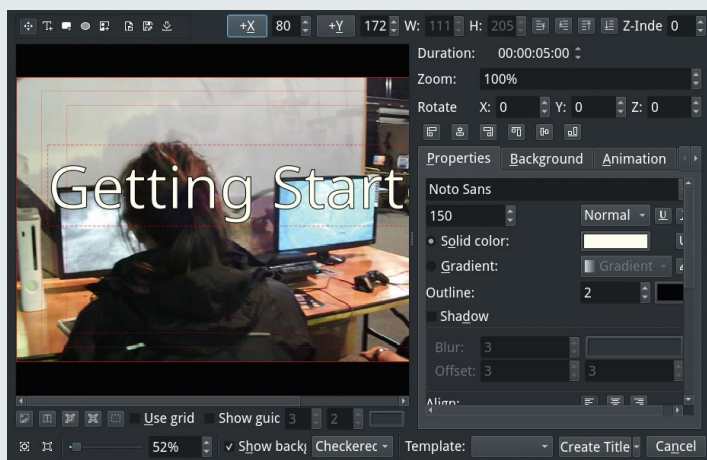
To resize a clip, hover over the end (front or back of the clip) and drag it to lengthen or shorten the clip. It's sometimes worth disabling the Snap feature while doing this (Timeline Toolbar) for extra precision. Having

» TITLE CLIPS

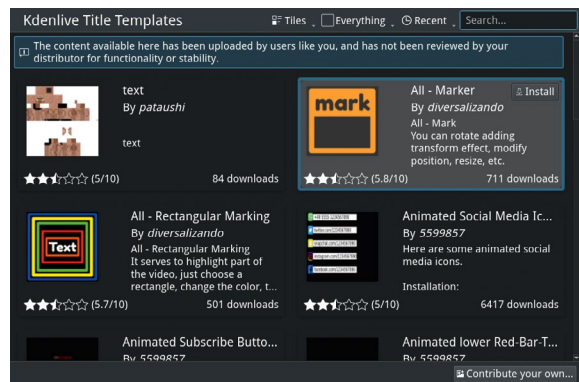
Title clips are used to overlay text on to your video, typically used to add an explanation for what's going on in the film or to introduce a section of a video. If you spend a lot of time coming up with something that you think you'll want to reuse, you can save a title card for use in other projects.

To begin editing a clip, simply select Add Title Clip in the Project menu – this creates the clip and launches the editor. The editor is WYSIWYG, and you can click anywhere to start adding text. Any parameters, such as the font size and colour, can be altered at any time. The default choice of white text with a black outline is a little bland, but it makes sense because it's legible on practically all backgrounds. If you click on the Show Background checkbox, you can preview what the title card will look like over the video under the Timeline Cursor.

There are also animation features (Animation tab), so you can have the text make an entrance from off screen. However, the best way to swish something up a notch is to explore the downloadable preset library that's hidden away and accessed via an icon in the toolbar.



The Title Clip Editor is quite complex but allows for some interesting ways of presenting text if you're willing to put in the time.



Kdenlive has an impressive collection of title templates with its own store-like interface complete with user ratings.

said this, on other occasions, the Snap feature is highly useful, because it takes into account the alignment with the starts and ends of clips on other tracks as well – perfect for accurate alignments.

There are two main approaches to cutting clips in *Kdenlive*. Firstly, you can select the Razor tool in the Timeline Toolbar. This enables you to cut a clip by moving the mouse pointer and clicking on a clip. The second method is to remain in Select mode to move the Timeline Cursor to the desired cut point. Press Shift+R to cut at that point. The advantage of this is that it's super-convenient to multiple-select clips (Shift+click) and then cut through them simultaneously. Generally, *Kdenlive* assumes you'll make the cuts in a rough way, then stretch or shrink the clip afterwards.

Transition talk

Quite often, you require a smooth fade between two clips to avoid a jump cut, and this is where transitions come in. In *Kdenlive*, a transition is a device that combines two clips in some way. Traditionally, the app didn't support clips that overlap each other on the same track, and a transition would have to be manually added between tracks that overlap.

Things have been made a bit easier in that department in recent versions, because you can now transition between clips on the same track. To do this, shorten two clips at the point at which they meet, and push them together. The shortening of the front and back of the clips is necessary so that there is some overlapping material to transition between. If you now double-click on the point where the two clips meet, a transition is created. For precision while doing this, it's usually a good idea to disable snapping.

The default transition is a wipe (a smooth crossfade), but it's worth exploring some of the other transitions, such as the slide, which moves the video from one clip over the other.

To choose a transition or edit its parameters, click on the transition and use the Effect/Composition panel to the right of the Timeline Editor. The parameters are often worth experimenting with. For example, the wipe can transition faster, or the slide can bring the second clip from the top of the frame rather than the default, which is to slide it across from the left of the frame.

Cause and effects

Kdenlive has quite a large palette of effects that can be dropped on to tracks, and they work in a similar way to

transitions, except that they work on a single clip rather than combining two clips.

To get started with effects, select the Effects tab in the rightmost panel area that normally displays the Project Bin. The effects are categorised; click on the little triangle to expand the category. From here, you can drop the effect directly on to a clip. This list is searchable, which is just as well as we wouldn't have thought to look for fades in the Motion category, for instance. There is also a small toolbar to filter the effects built into the panel, which is useful because the audio and video effects are all part of the same list.

In previous versions of *Kdenlive*, it was tricky to apply an effect to multiple clips on the same track, but this has been improved as you can now drop the effect on to the Master area of the track to affect the entire track – but you can still add effects to individual clips.

Whether adding effects to a clip or an entire track, effects are stackable; just keep dropping them on to the same destination. The only thing to watch out for is that effects are CPU-heavy, so adding too many can make the preview monitor sluggish and significantly slow down the rendering of the finished project.

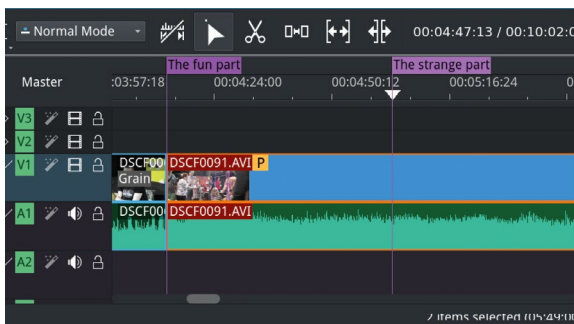
If you want to copy the entire effects stack from one clip to the other, use the right-click context menu to copy the clip and then use the Paste Effects option in the right-click context menu on the destination clip. This also works if you have used Shift+click to multiple-select more than one destination clip.

If you come up with some settings that you'd like to reuse in other projects, you can save the effect by clicking on the Save icon in the Effects Parameter panel. Access the effect again in other projects by clicking on the Show All Custom Effects icon.

Working with multiple sources

Kdenlive has a useful trick to sync up multiple audio and video clips. For example, you might record the video of a lecture by pointing the camera at the person speaking while capturing the audio with a recorder that is placed closer to the speaker. The task is therefore to sync the high-quality audio file with the video footage.

To synchronise these, start by dragging both clips on to the Timeline and placing them on separate tracks in rough alignment. Mute the audio in the video file (Mute icon on the Master section of the Timeline). Click on the separate audio clip and then right-click and select Set Audio Reference. There is some processing



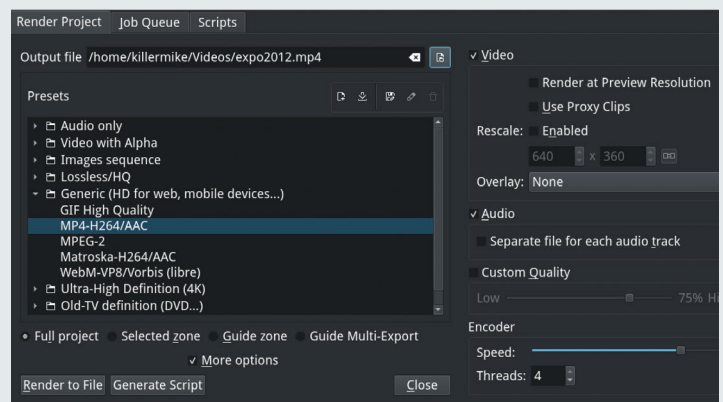
The Timeline can easily become messy and difficult to navigate, so consider dropping some guides (press the G key).

» RENDERING THE FINAL OUTPUT

Kdenlive offers many final output option presets, but a lot of these are for niche situations. In most cases, the preset Generic > MP4-H264/AAC is the right one. It's worth checking the More Options box to tweak things further, though. In here, don't be afraid to check Custom Quality and move the quality to 100%. This uses extra disk space, but it's better to overdo it with the quality settings, because if you intend to upload the video for online streaming on sites such as YouTube, there will be some quality loss as it is re-encoded.

Because video encoding takes a long time, make sure that the threads box has a number that is the same as the total number of cores on your CPU. Video encoding is a process that scales very well when split between CPU cores. That said, you may want to dial this back if you need to keep your computer responsive for other tasks while the encoding is taking place.

As is the case with the Title Editor, the toolbar has a link to a preset store browser. If nothing else, you can download these to examine when you're stuck for the right settings.



It's better to err on the side of overkill with output quality if you intend to upload the video to a streaming site.

that has to be done at this point as *Kdenlive* builds up a map of the peaks and troughs of the audio.

Once the processing is complete, it's possible to align the video footage with the audio clip as long as the audio is similar. In our example, click on the audio clip that is connected to the video clip, right-click and select Align Audio To Reference. The clip should shunt forward or back so that it's exactly aligned.

You can also use this to synchronise multiple video sources by using their audio tracks as the sync source. Imagine a situation where a musician is recording a tutorial video. They might have a close-up camera showing what their hands are doing, a zoomed-out camera showing their face and body, and an audio recording of their voice and their instrument. This gives them three source files to deal with. They can align all three perfectly by setting the audio clip as the reference track and using Align Audio To Reference to shift the two video files from the two cameras into alignment with each other and the audio track.

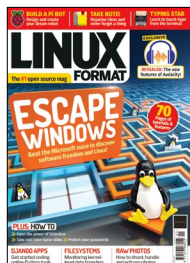
Kdenlive has all the basic features to make good-looking videos, along with a few options to add pizzazz to the finished product. Happy videography! **LXF**

» **PUT US ON YOUR OWN TIMELINE** Subscribe now at <http://bit.ly/LinuxFormat>

BACK ISSUES » MISSED ONE?

ISSUE 297 January 2023

Product code:
LXFDB0297

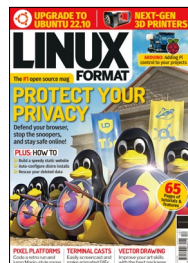


In the magazine

Escape from Windows by setting up and using the extremely user-friendly Elementary OS. We also help you take pro-level shots with our guide to RAW photography and present the inside scoop on the future of *Audacity*. There's a *Roundup* of the best note-taking outliner apps, too, plus tutorials on touch-typing, password management, building a Django app and a whole lot more.

ISSUE 296 December 2022

Product code:
LXFDB0296



In the magazine

Stay safe and protect your privacy online with our *Firefox* guide. Plus, discover what the future holds for 3D printing and how to upgrade to the latest version of Ubuntu 22.10 – Kinetic Kudu. Elsewhere in the mag we look at five vector drawing apps, review Raspberry Pi OS and have tutorials on recreating retro platform games, building static websites, creating a cataloguing app and much more.

ISSUE 295 November 2022

Product code:
LXFDB0295



In the magazine

Relive past computers and games with our in-depth look at retro emulation. We also round up five lightweight distros, step into the exciting world of Windows Subsystem for Linux, and chat to Alex Jones, director of Kubernetes Engineering at Canonical, about Kubernetes and the cloud-native future. There are tutorials on internet bandwidth, PC search and loads more, too.

ISSUE 294 October 2022

Product code:
LXFDB0294

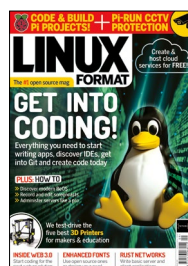


In the magazine

Get your system running the latest Mint, with other distros and even Windows! Elsewhere, we assess five creative writing tools, show how to keep the lights on in your smart home, build a Raspberry Pi media centre, turn smartphones into live cameras, perfect 3D prints, create better cut-outs in *GIMP* and recreate Pac-Man in Python. Plus we reveal the latest AlmaLinux developments.

ISSUE 293 September 2022

Product code:
LXFDB0293



In the magazine

We explain coding principles, what editor to use, how to use Git, and help you take your first steps writing your own code and controlling physical Pi projects. We review five of the best 3D printers, put together a CCTV system, assess a recreation of the BeOS operating system, and show how to create custom fonts. There's also tutorials on Web 3.0, screencasting and cloud hosting.

ISSUE 292 Summer 2022

Product code:
LXFDB0292



In the magazine

We demystify the world of graphics on Linux with our in-depth feature. Elsewhere, we edit photos with *Photoflare*, explore RISC OS, try out dev environments using Gitpod, control multiple LEDs around the home with a Raspberry Pi, and add gamepad support to *Tetris*. We also assess five software whiteboards for collaborative teaching, and talk to the talented coder behind the *Amiberry* emulator.

To order, visit **www.magazinesdirect.com**

Select **Single Issues** from the tab menu, then select **Linux Format**.

Or call the back issues hotline on **0330 333 1113**
or **+44 (0)330 333 1113** for overseas orders.

Quote the product code shown above and have your credit or debit card details ready.

USA? EU? THE MOON*?

UK readers
turn to
p16

SUBSCRIBE!

Don't wait for the latest issue to reach your local store – subscribe today and let *Linux Format* fly straight to you. Faster, cheaper and with DRM-free archive access!



3 GREAT WAYS TO SUBSCRIBE

Print, digital-only,
and print+digital
bundles!

» USA
From \$132
For 13 issues

» REST OF THE WORLD
From \$132
For 13 issues

» EUROPE
From €100
For 13 issues

IT'S EASY TO SUBSCRIBE!

Visit www.magazinesdirect.com/linux-format

Call +44 0330 333 1113

Lines open Monday-Friday, 9am-5pm, UK time.

*We don't actually deliver to the Moon. Yet.

More fun with layers and masks

Karsten Günther reveals how GIMP manages layers, what the different types are and how to use masks for subtle effects.



**OUR
EXPERT**

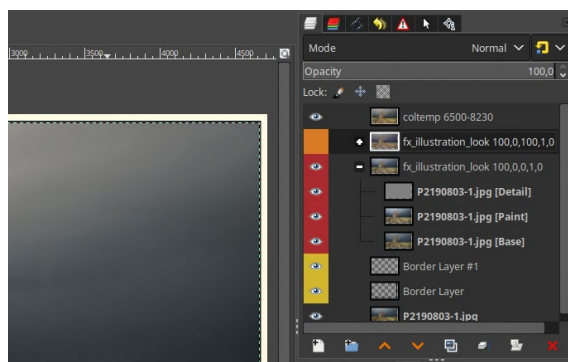
Karsten Günther is not only a master of using layers and masks to edit photos, but he also finds them useful to keep warm and incognito.

Besides selections, layers are one of the most important elements when creating complex images. *GIMP* is no different. Being familiar with layers also helps if there seems to be an error – often a different layer is active from the intended one.

Normally, we can think of layers as transparencies, slides or printouts stacked on top of each other. The viewer looks at the stack from above and sees what is still visible after covering it. In fact, layers are much more complex objects, as will become apparent.

The Layer Dock represents this stack of layers in a list. All layers present in an image appear there with their name – for transparent layers it is displayed in normal font, for opaque ones it's in bold – in the current order. All text layers, for example, are transparent, but other layers can be given this property by adding an alpha channel. If the eraser or cutting or deleting removes image material from such a layer, these areas become transparent. With non-transparent layers, the current background colour appears. Transparent layers do not necessarily need to have transparent areas.

GIMP uses layers for various purposes. Special parts of an image, such as text, are automatically



Lots of layers in one image are not rare but the rule. The Layer dock shows the connections and enables you to adjust multiple aspects.

created on a separate layer. It is still common practice to duplicate layers before editing, such as blurring – as a backup – in order to be able to access the original image data at a later time. Then an area of the image is processed, possibly in different variants, which are temporarily stored on different layers.

Layers are therefore mainly used to edit parts of an image. This is true even if an entire layer is duplicated

» LAYER MODES

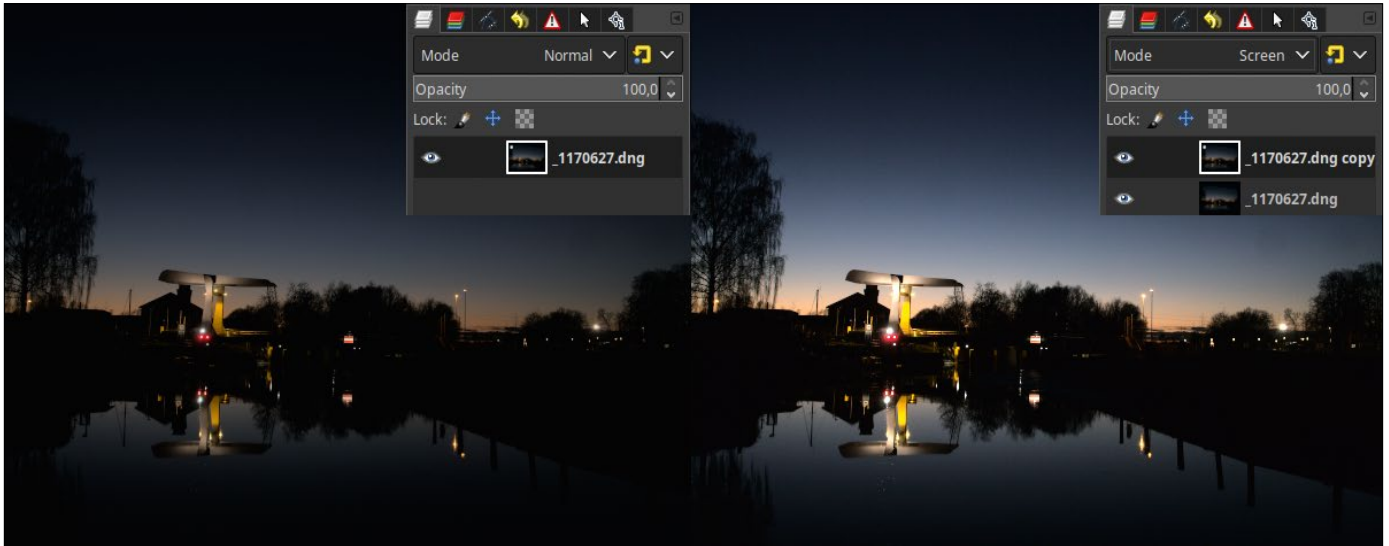
Layer modes are essentially an offsetting or rendering rule of the RGB values of the pixels in two neighbouring visible layers. The layer mode of the upper layer determines how the pixel values are offset against the layer below. *GIMP* then displays the result of this calculation. The outcome often looks rather strange, but it can be extremely useful. For example, the positions of layers can be checked to make sure they are pixel perfect with the Grain Extract mode, where any deviations clearly stand out. The subject is covered in detail in the *GIMP* manual at https://docs.gimp.org/2.10/en_GB/gimp-concepts-layer-modes.html, including the formulas used for offsetting.

In practice, it is often helpful to place your mouse cursor over the Mode button and then use the mouse wheel to try out all the different modes in

turn. However, a lot of modes have very strong effects, so they tend to only be used on layers that have reduced opacity.

A typical application of layer modes is to darken or lighten highlights and shadows. To lighten dark images, for example, the Screen mode is a good choice. It has the effect that dark areas are lightened much more than light areas, which corresponds to how the human eye perceives things. You can see *GIMP*'s Screen layer mode in action in the screenshot (above-right). In some situations, the Addition and Dodge modes can also be helpful. Several copies of a layer can be superimposed to intensify the effect.

For images that are too bright, the Overlay and Multiplication layer modes, along with the different Light modes (Soft, Hard, Vivid), should be tried.



and offset against the copy via a layer mode. Layer modes have no counterpart in the physical world; they are part of the magic of image editing.

When using a lot of layers, *GIMP* gives you a hand with several helpful features. For instance, layer groups are containers that can contain any number of layers. They can be displayed together in the Layer Dock, which increases your overview immensely. The layers in the group can still be edited individually and their visibility can also be controlled on an individual basis. In a group, all layers become invisible if the visibility of the layer group is deactivated.

The colour markers for layers are still a relatively new feature (v2.10). Applied via Layer Attributes they can help with orientation and be set however you like.

Layers can be moved freely in an image, even if they are as large as the current canvas or larger. The canvas size automatically increases accordingly. For moving layers, *GIMP* offers two tools in the toolbox: the move tool – accessed via the *m* key as a shortcut – and alignment, for automatically aligning lots of layers. The move tool works well with magnetic guides (Snap To Guide) or with a magnetic grid (Snap To Grid), both in

the View menu. This way, several layers are positioned exactly and do not appear to be shifted by a few pixels.

The position of layers is usually of great importance. If they are shifted by only a few pixels, it can have a very visible effect. *GIMP* therefore allows you to fix the position of a layer – just click on the moving symbol in the Layer Dock for the active layer in the Lock line.

Sometimes it is not the absolute position of layers that is important but their relative position to each other. For example, text should often be placed directly above or next to an inserted object. In this case, several layers can be linked together. In the Layer Dock, this is done via the chain symbol next to the visibility eye. On linked layers, *GIMP* applies the same tools with the same settings one after the other.

The visibility of layers must not be confused with their opacity and other aspects. The eye on the left side of the Layer Dock shows whether each separate layer is currently visible. This is independent of whether a layer is currently active – ie can be edited. In *GIMP*, only one layer is ever active. *GIMP* shows which layer is active in the Layer Dock with a colour bar. Basically, all actions carried out with *GIMP* always refer to this

In Screen layer mode, *GIMP* lightens dark areas of identical layers more than light ones. On the left is the original layer; on the right Screen layer mode is used.

QUICK TIP

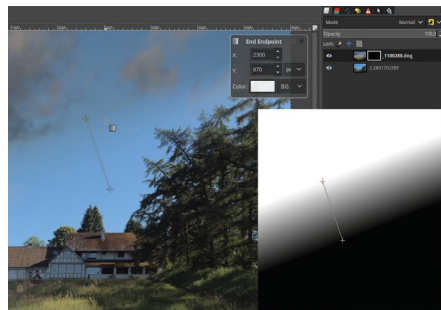
Wikipedia explains the historical development, background and ideas behind the alpha channels quite well (https://en.wikipedia.org/wiki/alpha_compositing).

ADD DRAMA WITH A SOFT LAYER MASK



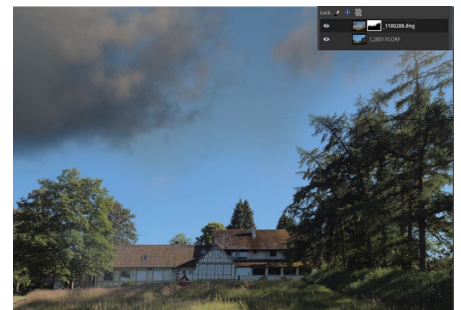
1 Starting point

The sky of the upper layer is given a mask so it only covers the parts of the sky that appear too uniform on the image below. If the layer mask is initially filled with black, the layer above is initially invisible, which is advantageous in this example.



2 Use a gradient

Click the layer mask icon to activate it for further editing (although it remains invisible). The gradient tool creates a colour gradient from black (invisible) to white (fully visible) on the mask in any direction. The preview shows the effect.



3 Make corrections

The tool still allows you to move the end points and even to create a base point within the gradient. Now, using a soft brush, small corrections can improve the mask further. Holes in the mask can be closed and the edges can be edited.



QUICK TIP

Text layers are special objects in GIMP. In the dock, they are marked by the letter T. Their content can be changed until a graphic operation converts the layers into normal transparent image layers.

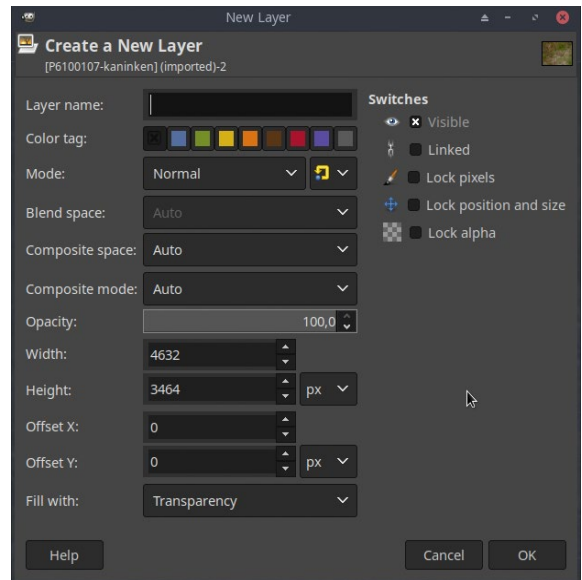
active layer. If it is not visible at the moment, *GIMP* warns you. However, that only helps to a limited extent; if another layer covers the active one, the effects of the actions are often still not visible.

The other attributes a layer can have are shown in the now very extensive dialog for creating a new layer. All attributes can be changed later, should this become necessary. The corresponding functions can be found in the context menu of the layer in the Layer Dock.

GIMP is quite efficient with layers: they are always rectangular and many tools create layers with a minimum size, so that they are at least partially filled with pixels at the edges. However, layers can be adjusted to the current image size or any other dimension at any time.

All functions for editing layers can be found in several places: in the normal menu under Layers and – often more quickly accessible – in the context menu of the Layer Dock, as well as being coupled to the buttons that are available under the layer list.

One long-standing special peculiarity of *GIMP* is the so-called floating selection. It always occurs when material saved from the clipboard is inserted into an image. Many people find floating selections to be particularly annoying, because they refer all actions to themselves as long as they exist. So, if the key



When creating new layers, a number of special settings can be made. The online help explains the details, including composite and blend features: https://docs.gimp.org/2.10/en_GB/gimp-layer-new.html.

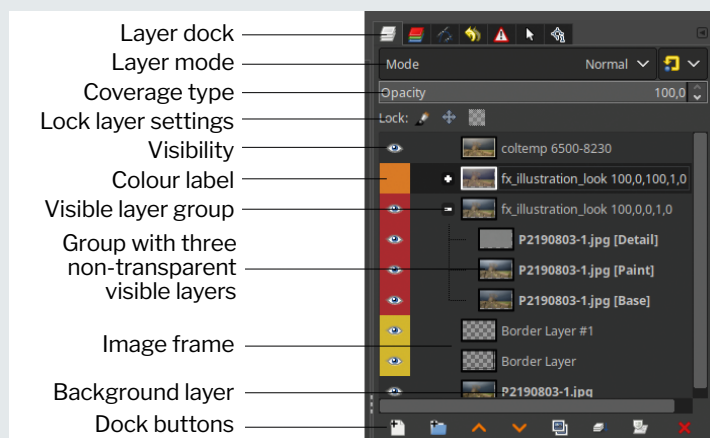
combination Ctrl+v inserts a floating selection, it must be converted into a normal layer – for example, by the New Layer button under the layer list in the dock – if another layer is to be activated for editing. However, there is another way: layers inserted in this way are often used as patches. Therefore, a floating selection can be moved directly with the mouse, and at the correct position a mouse click near this layer anchors it to the layer below.

» THE LAYER DOCK

The linchpin for all layer operations is the Layer Dock. There, layers can be created, rearranged, deleted, named, marked, joined and more. With *GIMP* 2.10 the Layer Dock has become quite complex. Images can contain many layers, which are then grouped together and often only partially visible. Transparent layers (layer name in normal font) and opaque layers (layer name in bold) can be mixed and, if necessary, converted into each other with the alpha channel.

In the Lock section, *GIMP* allows individual aspects of layers to be specifically locked. In this way, layers can be protected from being painted over, moved or, in the case of transparent layers, having changes made to the alpha channel.

The most important general functions in the Layer Dock are linked to the buttons below the layer list. Functions for a selected layer can be found in the context menu. The Layer effects plugin also adds its submenus for the individual effects at the end of the context menu in the Layer Dock.



The use of several layers in the layer dock is nowadays facilitated by colour marks, layer groups and so on.

Layer masks

Masks, while analogous to alpha channels in many ways – making parts of layers visible or invisible – are completely independent of them. Like an alpha channel, a mask is a black-and-white representation, where the brightness represents the visibility of the masked pixels. Pixels that have a white background in the mask appear with maximum visibility, while pixels masked in black are invisible, and greyscale is interpreted by *GIMP* as attenuated visibility.

These properties make layer masks particularly useful for three applications:

- Cropping or cutting out subjects, as described in the *GIMP* tutorial in LFX297.
- Colourkey renderings, which are in some ways a special variant of this.
- Creating smooth transitions between two layers with a gradient from black to white.

In the Layer Dock, *GIMP* shows layer masks as an additional icon to the right of the layer icon. In an active layer, either the layer itself or its mask is always selected for editing. *GIMP* indicates this by a white frame around the corresponding icon. (This is often not easy to see with masks.) A mouse click on one of the icons then activates either the layer or the mask.

Masks can be edited manually if necessary. As long as the layer mask is active, it can be edited like a normal black-and-white layer. Normally this is done with a soft brush, but other tools and filters can also be used. Often the Show Layer Mask function (procedure name: `layer-mask-show` in the settings) can be helpful

here. It can be useful to bind it directly to a shortcut in order to be able to switch it at any time. Important: the layer mask must be activated in the dock in order to edit it.

Using masks in combination with a gradient is often a good choice because it requires little post-processing. However, you can also try to create a mask that is as exact as possible. As an example, this can be done as follows: the foreground layer used as a mask is edited with the Curves and/or Threshold tool so that necessary parts appear black and unnecessary parts appear white.

Separate colours

In some situations, it is important to be able to process the colour components in an image separately. This may be the case when preparing an image for printing or, for example, to avoid artefacts when sharpening by processing only the brightness component of the image. The separation of colours into their components is often called colour separation – in *GIMP*, the term decompose is used. In the colour menu, under Compose, there is a corresponding tool for this task.

The application is simple and works on the current layer: after the call, *GIMP* shows a simple dialog. Here, it is only a question of which components are to be created. The components are created by direct conversion of the RGB pixel values and are output in the form of three to four layers in a new image. The colour model set in the dialog determines which



Two layers made this, the upper one desaturated and with a mask. The colour shines through from the bottom one.

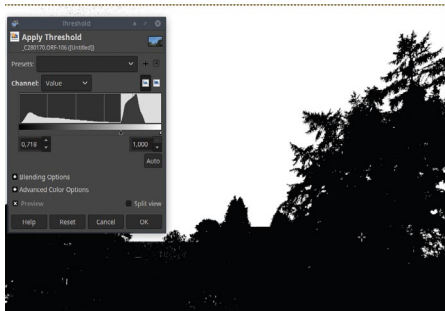
components *GIMP* creates and how. Each of the generated colour components is a greyscale image of the original layer; all together, they generate it again.

Colour models are complex mathematical systems that are described in detail in Wikipedia. The example above uses LAB (or actually $L^*A^*B^*$), an established colour model that stores brightnesses in the L component. The new image created with the dialog now contains the three layers L, A and B, provided the option Decompose To Layers was active. All the layers are pure greyscale images. They can be edited separately. Compose puts them together again afterwards. *GIMP* also automatically updates the original layer. **LXF**

QUICK TIP

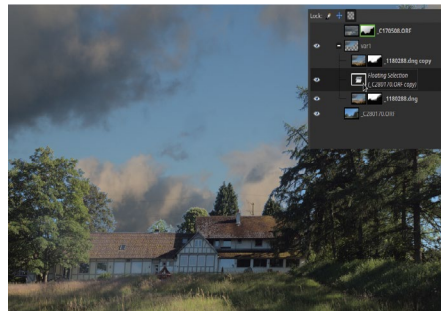
There are some interesting functions, such as inverting the layer stack, minimising layers and so on, which are only listed in the Layer menu.

USE A HARD LAYER MASK



1 Threshold mask

You can also create a mask that is as exact as possible, which hides the areas that are to be replaced (in the background). In this case, the lower layer contains the sky and has no mask. The upper layer contains the foreground and is masked black where the sky should appear from the background. Here, the Threshold tool can create a mask from a copy of the foreground layer. The mask generated in this way only needs to be slightly adjusted.



2 Insert the mask

If the layer mask is active, any layers can be copied into the mask. This is how it works: the desired layer for the mask is first copied to the clipboard with Ctrl+c, then a mouse click activates the layer mask selected as the target. Ctrl+v creates a floating selection in the dock, which is displayed above the selected layer. A mouse click on the anchor symbol appearing in the button bar anchors the floating selection as a mask.



3 Correcting artefacts

The requirements for a hard mask are high – for instance, it may be necessary to rework the edges a little if artefacts appear in the layer. The Erode and Dilate filters from the Generic submenu sometimes help here, but never forget you can directly draw and manipulate the mask channel as any normal image. The Warp tool can be very useful here as you can “nudge” parts of the mask edges into position more easily than drawing them.

» **GET LAYERS AND LAYERS OF LXF!** Subscribe now at <http://bit.ly/LinuxFormat>

Create better QR codes

Mike Bedford reveals how to get creative with your QR codes to draw attention to them better and help people sign up to newsletters and more.



**OUR
EXPERT**

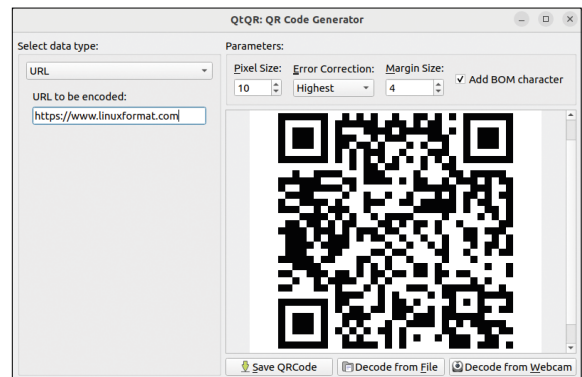
Mike Bedford has long been fascinated by QR codes, and has been inspired by their resurgence in popularity to create less boring-looking variants.

QR codes, those two-dimensional barcodes that whisk you off to a web page when you scan them with your phone, have had a chequered history. Invented in the '90s and used extensively in Japan, their belated introduction to the rest of the world was short-lived before heading for apparent extinction. You'll probably have noticed, though, that they're now enjoying a resurgence in popularity. Fuelled by their use during the Covid pandemic – remember having to scan QR codes to enter restaurants and pubs? – we're now seeing them everywhere as everyone has been trained up to use them.

If you own a business, putting a code on your business card or brochure makes it far easier for potential clients to access your website. If you edit a newsletter for your sports team, code club or local community, QR codes can engage your readers better than a printed URL. What's more, some publications allow images in their small advertisement columns, so using QR codes there is another option. But here's an interesting thing: recent surveys have shown – hardly surprisingly – that if you move away from a boring black-and-white matrix to something more creative, you'll get far more hits. So, the subject of this tutorial is how to create artistic QR codes.

The basic code

Our approach for creating QR codes that stand out from the crowd is to create an ordinary QR code and



The first step in our approach to making a creative QR code involves making a regular QR code. We suggest using QtQR.

edit it. Because while there are plenty of tools for generating basic ordinary QR codes, we don't know of any Linux utilities for producing artistic variants, but we do have some good image-editing tools, such as **GIMP**, which we're using here.

To create the basic code, we're almost tempted to say that a GUI tool offers little benefit over a command-line tool. After all, typing `qrencode LXF -l H -o https://linuxformat.com` is no more of a chore than typing the same information and selecting the odd menu entry in a GUI tool. However, you're not done at that point because, after you've generated the QR code, you're undoubtedly going to want to take a look at it, so you need to issue a command to another utility, which probably isn't installed by default. So, we're going to suggest you use the GUI tool **QtQR**, which displays codes as soon as they're generated.

Our main theme here is how to augment basic QR codes in **GIMP**, so we're leaving you to figure out how to use **QtQR** to produce the basic code, but it's not at all difficult. There's a couple of points we ought to make, though. First, select the highest error correction level, sometimes referred to as just H, because our subsequent modifications will make the code more difficult to read. And second, if you're going to customise it by adding an image, create a large QR code in terms of pixels, otherwise your image has to be very low resolution. Alternatively, resample the QR code to a higher resolution using **GIMP**'s Image > Scale Image option before you use it. That works perfectly

» ALTERNATIVE APPROACHES

We'll have to leave you to figure out the details yourself, but here's a couple of other ideas you might like to try. First, the black squares don't all have to be the same colour. Using a rainbow of colours might look gaudy, so we have a suggestion that's more subtle and more likely to be readable. Use an image that comprises a fountain fill or similar pattern, in a range of similar darkish colours, imported as a layer in **GIMP** below the QR code, and make the black areas of the QR code transparent. Second, so long as you don't overdo it, the eyes can be made a different shape and this slight change can look quite effective. Circles, or squares rounded off on one or all corners, have been used effectively, but the overall size shouldn't change, and they should still be a central shape surrounded by a border.

with a QR code, so long as you choose None for Interpolation.

Squares and eyes

We're going to look at several ways of brightening up a QR code and you can use these either singly or in combination. However, bear in mind that the more changes you make, the less likely it'll decode correctly. So, be sure to test your modified code before using it in earnest, ideally using several different phones and scanning apps. And finally, before looking at specific techniques, because all our methods involve introducing colour, after you've opened the QR code in *GIMP*, ensure it's a colour image by choosing Image > Mode > RGB.

The simplest – but still quite effective – way of brightening up a QR code is to colour the black squares. This is, perhaps, safer than colouring the background (the white squares). The methods in this section can make quite a difference alone, but if you also want to add an image, as discussed in the following section, we suggest you recolour the QR code first.

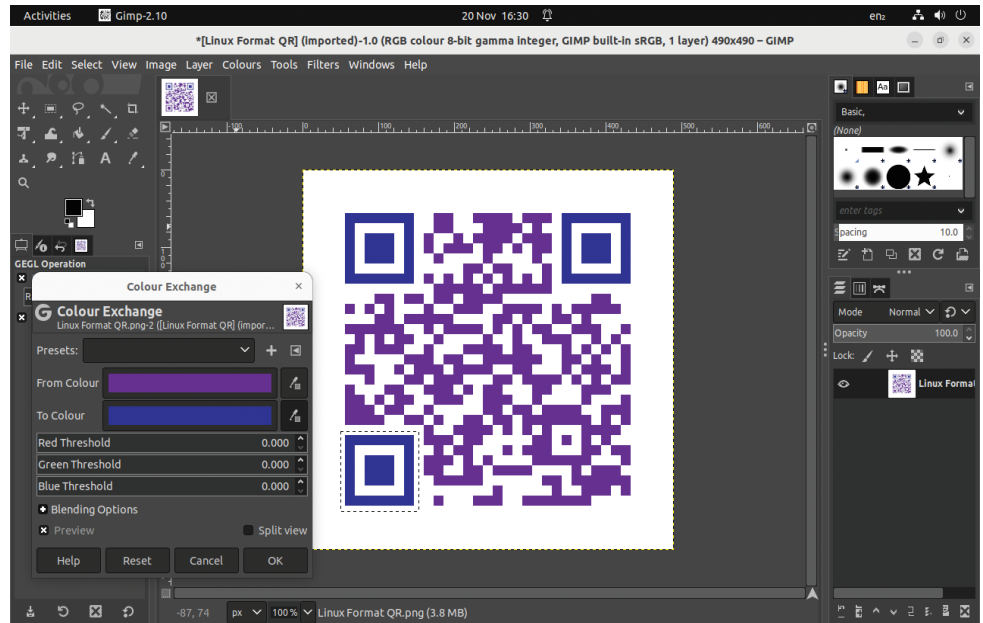
With the QR code open in *GIMP*, select Colors > Map > Color Exchange. Now, in the Color Exchange dialog box, select black as the From Colour, and your selected new colour, which should be fairly dark, as the To Colour. All the black content of the QR code changes to your chosen colour so, instead of black squares on a white background, you'll have more colourful squares.

As well as colourising the black squares, using Color Exchange also alters the colour of the so-called eyes, the square patterns in the top-left, top-right and bottom-left corners. Artistic QR codes sometimes have differently coloured eyes, but this needs us to change just part of the QR code to a different hue. To do that, first select one of the eyes using the Rectangle Select tool. You can now use the method we used to recolour the squares, but this time it affects only the selected area. Repeat this for the other two eyes and you're done. You could also vary this method slightly to use different colours for the central square of the eyes and their border.

Add images

Next are two different ways of adding an image to a QR code. In both cases, our instructions assume the QR code is open in *GIMP* and you've made sure the image mode is RGB.

First up, we're going to see how to incorporate a small image in the centre, this being one of the most common ways of customising a QR code. The image must be fairly small,



■ GIMP provides so many options for enhancing a boring monochrome QR code; changing colours couldn't be much simpler.

perhaps replacing about 100 squares, and it should have a white background. The technique is quite straightforward. Open the image you want to add as a new layer (File > Open As Layers) and ensure the image is the top layer. Now scale it, using Layer > Scale Layer. It looks better if you size it so no partial black squares are visible around it.

It's not as common, but a different way of adding an image, which you might like to try, is to add that image as a background behind all the black squares of the QR code. However, there's a lot more scope for things to go wrong here, because that image will be the full size of the code. An image in a few pale colours, say yellow and light cyan, might work OK, but with a photographic image, you first need to increase its brightness so all the colours will be much fainter than the squares of the QR code itself.

Open the image as a new layer, but this time it should be the bottom layer. Resize it, if necessary, so it's about the same size as the code. With the image layer selected, chose Layer > Transparency > Color To Alpha, and in the Color To Alpha dialog box, ensure that the default colour white is selected. The image now shows through the QR code wherever it was white. **LXF**

QUICK TIP

A QR code with the highest error correction level (H) can be read if 30% of it is damaged. However, don't push it by placing too large an image in the centre, because that means it's much less immune from accidental damage or poor lighting.



Using a small image in the centre of a QR code is common and effective, but you might like to try a full-size but faint background image.

» **WE COULD USE A QR CODE HERE** Subscribe now at <http://bit.ly/LinuxFormat>

OBSIDIAN

Credit: <https://obsidian.md>

Nifty note-taking with Obsidian

Nate Drake walks you through adopting the almighty Markdown-based note-taking app Obsidian to see whether it lives up to its legend.



OUR EXPERT

Nate Drake is a tech journalist specialising in cybersecurity and retro tech. Using Obsidian to plan out his project for lighthouses has helped him discover that they're actually very heavy.

Obsidian's website describes the app as "a second brain for you, forever". In plain English, that means a tool that can help you take and organise a colossal number of notes in a simple way, from basic study notes to a full-blown wiki.

The program makes use of the Markdown markup language to format text, making it easy to import and export files. This simplifies tasks such as using links to jump between notes and organising large amounts of information.

Obsidian is not FOSS but the free tier fully supports Linux, doesn't require registration and has some very powerful features. The program itself was developed by Shida Li and Erica Xu, who were unhappy with the existing note-taking software on offer.

There's also a thriving Obsidian community (<https://obsidian.md/community>) and a host of devs creating plugins to improve its rather minimalist interface.

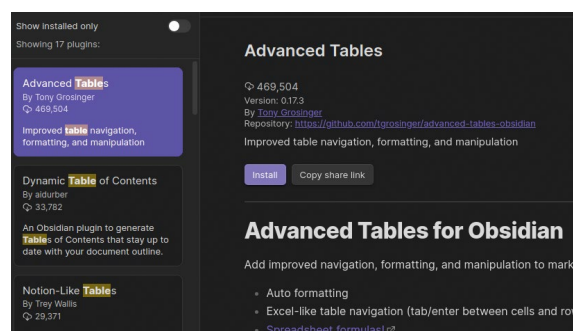
Pick your package

Obsidian is extremely Linux-friendly, being available via Snap, Flatpak and an AppImage. Head over to <https://obsidian.md/download> to get started.

Ubuntu users will find it easiest just to click Linux (Snap) to download the file. Open the terminal and use `cd` to navigate to your **Downloads** folder. You can then use `snap` on the relevant file to install it – for example:

```
$ snap install --dangerous --classic obsidian_1.0.3_amd64.snap
```

Note that because this is a third-party Snap file, you must use the `--dangerous` flag. You also need to enter



Expand Obsidian's features by installing free community plugins, such as Advanced Tables, which supports inserting Excel-like sheets.

your admin password. Once installed, *Obsidian* appears in Applications.

The *Obsidian* website, however, uses a large purple button to push the AppImage option, which has the advantage of working with virtually every flavour of Linux. The AppImage can also run directly from the folder where it's downloaded.

If you want to install this way, first make sure you have both **fuse** and **libfuse2** by running `sudo apt-get install fuse libfuse2` from the terminal. Next, right-click the file in your **Downloads** folder. Choose Properties and toggle Executable As Program. (Hardcore Linux users may prefer to use `chmod +x` to do this.) Next double-click to run *Obsidian*.

The installers weigh in well under 100MB, making *Obsidian* extremely lightweight.

One from the vaults

On your first launch, *Obsidian* offers three main options. Resist the urge to do a Quick Start and click Create to open a new so-called vault.

You first need to set a name for the vault, which will contain all the Markdown files for this particular project. Next, click Browse to choose where to save it. *Obsidian* automatically creates a folder with the project name – **Lighthouse Project**, for example – but we took the opportunity to create a parent folder for this and future vaults, imaginatively named **vaults**. Click Open, then Create to confirm your choices.

When the *Obsidian* dashboard itself launches, you'll see the interface is very spartan. This is deliberate and

QUICK TIP

If you haven't used Markdown before, check out Matt Cone's awesome guide at www.markdownguide.org. Both the Basic Syntax and Cheat Sheet pages have all the information you need to get started.



Obsidian uses Markdown to easily link between different notes and images. The program suggests media in your vault as you type.

in keeping with the Markdown philosophy of having a simple, stripped-down text interface. On the plus side, it's easy to navigate.

Start your first *Obsidian* Markdown note by clicking Create New File. We decided to begin with a **Statement of Intent** for our project. Change Untitled to give this note a name.

From here, you can enter text in Markdown format. If this is your first time using Markdown, we should explain that it's extremely easy to pick up. *Obsidian* has already created a main heading for you. You can also create subheadings by placing **##** before a new line of text. Use one or two ***** before and after text to make it italic or bold respectively.

You can also place web links in Markdown documents – for example:

My favourite magazine is [Linux Format](https://linuxformat.com/).

There are several ways to write lists. The simplest is to number them with a bracket – for example, **1)**, **2)**, **3)** and so on – because *Obsidian* automatically recognises these and formats your text accordingly. Place a **---** to draw a neat line.

One particularly impressive aspect of *Obsidian* is that you can also paste in formatted text and it does its best to preserve the formatting. In our tests, we pasted in a paragraph from Wikipedia and all the special characters and web links were preserved. This may be a relief if you have existing documents that you want to use with *Obsidian*.

It's clear that *Obsidian*'s developers have tried to make the editor as WYSIWYG as possible, although you can click the icon at the top-right to switch between reading and editing mode. Markdown purists may prefer to click the ... options icon to switch to Source Mode to see the raw text.

Click the New Note icon at the top-left each time you want to create a separate file. Alternatively, click + to open a new tab and create or open a note from there to switch between them easily. All notes are saved automatically in Markdown format to your vault.

Pics and links

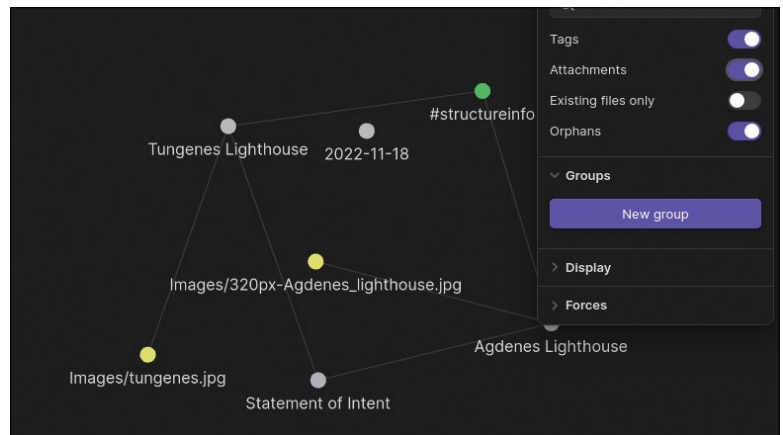
Markdown was built to allow easy cross-linking and sharing of info, and *Obsidian* has turned this into an art form. The easiest way to place an image inside a note is firstly to download it to the same folder as your vault.

Then, when inside a note, just type **![[** to open a small menu and select your image. Click to embed the image in your note.

There may be times where you want to link to another note or picture without embedding the whole thing in the file. For instance, our **Statement of Intent** has links to other notes containing more information about the lighthouses we plan to restore. If you want just to include a link, simply type **[[** without the **!**.

If you're managing a large project, navigating all these links and pictures may not be easy. To make life simpler, consider clicking the New Folder icon at the top-left to drag and drop pictures or notes into logical categories, such as images. *Obsidian* automatically updates any existing links.

For a full top-down oversight of your project, click Open Graph View in the left-hand pane. This is where *Obsidian* really comes into its own, because you can



see the various links between notes and even group them in more advanced ways.

One good way to better organise your notes is by using tags. Tagging a note is just a matter of writing **#** anywhere followed by your tag – for example, **#historynotes**. Click Open Graph Settings on the right while in Graph view to toggle displaying tags.

The graph view is an excellent feature for seeing all your notes in one place and how they relate to each other.

Observations

There's much to love about *Obsidian*. Setup is fairly easy and writing notes is a dream, even if you're not very familiar with Markdown. This article has focused on the free version of the app, which doesn't include the billable Sync and Publish services.

Sync enables you to save your notes in encrypted format across all your devices, though for only slightly more trouble, you can install a plugin to encrypt your notes (see *boxout*, below) and save them to a cloud folder. *Obsidian* supports exporting notes to PDF format and Markdown plays well with most word processors, but Publish is probably the best way to go if you want to transform your notes into a website.

If, however, you're happy putting time into learning Markdown and with taking notes on one device, *Obsidian* is very powerful and easy to navigate. **LXF**

»» OBSIDIAN AUGMENTATION

Obsidian is designed to be a stripped-down note-taking app, ideal for importing existing projects or starting new ones. You can hugely expand its functionality, though, by using plugins.

To get started, you need to manually enable community plugins. Click the Gear icon on the bottom-left and choose Community Plugins > Turn On Community Plugins to leave Restricted Mode. Next, click Browse to see what's available. Search or scroll to find a plugin you like and click Open In Obsidian. *Firefox* asks you to confirm by clicking Open Link. Now click Install. Once installed, you may need to click Enable to get it working.

We especially like the Day And Night theme, which lets *Obsidian* automatically toggle between modes, and Advanced Tables, which makes drawing tables much easier than typing them in Markdown.

The Calendar plugin allows *Obsidian* to display a simple calendar in the right-hand pane. (You need to click the Expand icon at the top-right to see this.) Click on a day to view your existing to-do list for that date or to create one.

Make sure you back up all your work first before installing plugins. You may need to restart *Obsidian* for them to take effect.

SERENITYOS

Credit: <https://serenityos.org>

Get Serenity – direct from the source

What do you get if you take a modern kernel, a classic GUI and a handful of custom apps? **Michael Reed** investigates SerenityOS.



**OUR
EXPERT**

Michael Reed was already serene enough, even before he tried SerenityOS, thanks to his experience with alternative OSes.

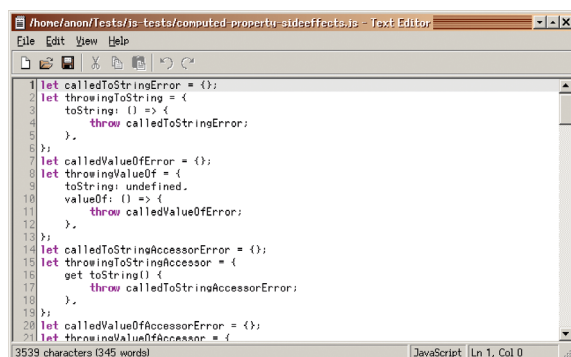
SerenityOS is an alternative, free and open source operating system that combines a modern technical approach with a healthy nod to classic operating systems of the past. The kernel has features such as pre-emptive multitasking, memory protection and multithreading, and overall, the system is quite Unix-like, with many familiar tools available from the command line. It takes its visual style directly from '90s classic desktops from the likes of Microsoft and Apple – a welcome sight to anyone who's fatigued by the bland, flat look of modern GUIs.

In some ways, it's fairly early in its development, but it already has a full desktop with associated tools and some runnable applications. One of the most admirable aspects of project is its highly active community, many of whom contribute to the code. This community has also funded the project, meaning that the main developer, Andreas Kling, is able to work on it full-time.

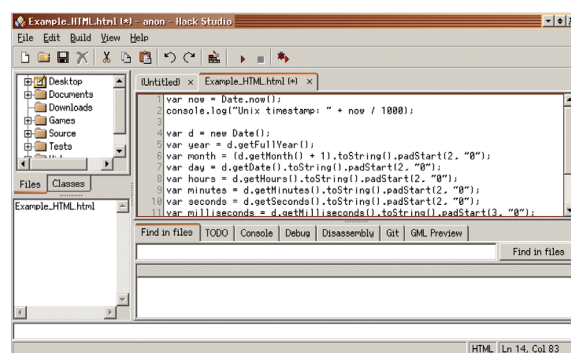
This all adds up to an alternative operating system that is developing at a mile a minute, and it's already good fun to play around with. To get the behind-the-scenes picture of what's going on with the project, we managed to shoot a few questions at Andreas himself.

Serenity from the source

As always, we'll start our exploration of the system by obtaining it and installing it. At the present time, there is no ISO and no traditional installer. The instructions for running SerenityOS involve cloning the Git repository (<https://github.com/SerenityOS/serenity>)



The text editor is a good example of the SerenityOS approach because it combines classic looks with modern features.



The IDE Hack Studio is an example of the ambitious tools that the community is working on for SerenityOS.

and running two scripts. The second script compiles the code of the OS and runs it within the *Qemu* virtualiser. If you're not familiar with Git, this might sound complicated, but it's actually easy to get up and running. The only downside is that compiling the system can easily take an hour on slower hardware.

When the script has completed its work, it launches a properly configured *Qemu* instance, and it's time to start having some fun with the system.

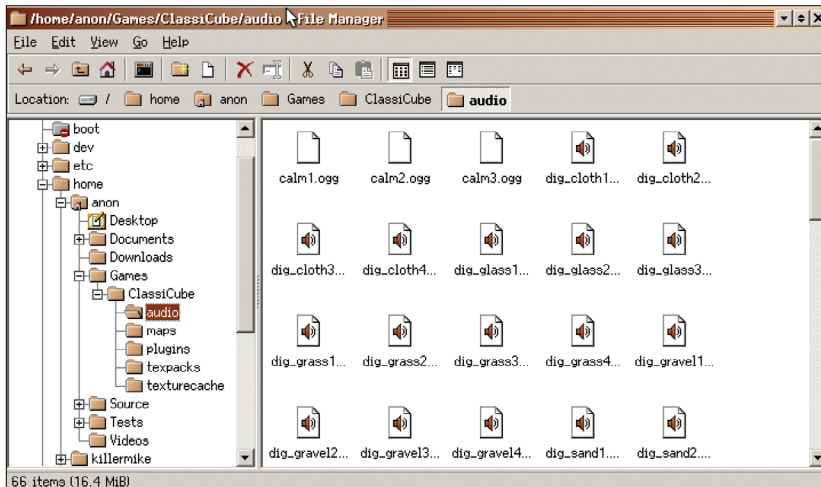
GUI and CLI

When it comes to the graphical style of the user interface, it's obvious the developers are big fans of a certain Microsoft OS that came out in 1995. We put the point to Andreas. "Absolutely! The first OS I ever used was MS-DOS, but I've always felt that Microsoft reached peak GUI around Windows 2000. After that point, the industry became distracted with consumer computing and professional, productivity-focused design became a niche rather than the norm. In my mind, SerenityOS is a return to that paradigm, and an exploration of what comes next if we keep ourselves (ie power users) as the target demographic."

Aesthetically, the Windows 95 vibes are strong, but many of the GUI support tools also look similar to Windows ones. For example, the network configuration tool looks like that of Windows. Many of these older-style configuration tools are starting to be pushed out of sight on the modern Windows desktop, and seeing them here fits in perfectly with the ethos of

QUICK TIP

For the moment, the default user is called 'anon' and the default password is 'foo'. Occasionally, you are prompted for this password when you invoke the GUI settings tools. If you want to become the superuser at the command line, use the 'su' command, in the same way you would on Linux.



File Manager looks suspiciously similar to the Windows Explorer of older versions of the Microsoft operating system. Fortunately, it's a good design.

SerenityOS. It's clear that the developers are aiming to take features that they admire from other systems.

Stepping away from the graphical environment, much of the command-line environment will be familiar to Linux users in terms of how it works. You list the contents of the current directory with `ls` and change directories with `cd`. You can list current processes by listing the contents of the `/proc/` directory. Despite the GUI looking Windows-like, we would have been disappointed if the command line was DOS-like.

SerenityOS has a working TCP/IP stack, and running standard network tools such as `ping` works as expected. It comes with a simple web server that's an original piece of software created for SerenityOS.

Applications and more

SerenityOS already has a reasonable amount of runnable software. It can only run its own software, but this includes some that has been ported from other systems. Looking at apps, most of what's available for it either falls into the category of GUI software that has been created by the SerenityOS community or ports of mostly text-mode applications and utilities. We were happy to see text editor *Nano* in this list and some will be pleased that there is a *Vim* port. There are also ports of some games, such as *Doom* and *Quake*.

Speaking of ports, for the moment, the package installation method is rather low tech, but it gets the job done. Software installation is carried out outside of SerenityOS. One navigates to the appropriate

subdirectory within the source code tree and runs the **package.sh** script contained within. This script downloads any needed files, compiles the source code and installs the packages on the SerenityOS disk image. It's not pretty, but it works for the most part.

The native SerenityOS GUI apps include things like the system text editor (*Text Edit*). This is fairly complete within the lightweight category, and includes features such as syntax highlighting. There is also a complete IDE called *Hack Studio*.

Software development is obviously a priority for the

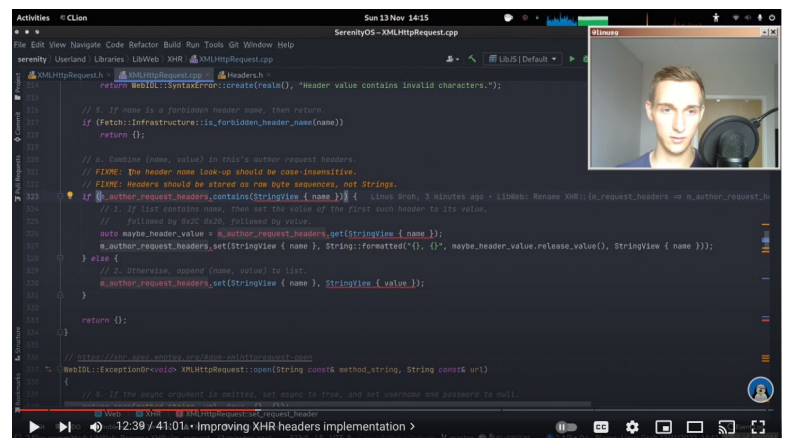
SerenityOS community because, as well as the IDE, they are developing a programming language called Jakt. This translates into C++, which it resembles, and it aims to have robust memory safety features like Rust. We could wonder if there is a danger that the SerenityOS community is spreading itself too thin, but we'll assume it knows what it is doing as it is producing results at an impressive pace.

A ladybird's web?

Ladybird, SerenityOS's own web browser, is the largest single SerenityOS app. We took this for a spin, and it worked reasonably well. An excellent web browsing

QUICK TIP

To update SerenityOS, type `git pull` followed by `Meta/serenity.sh run` while in the source code directory. Put the launch command into a text file and use `chmod +x [name of script]` to make it executable to save memorising the command sequence.



The YouTube channel of Linus Groh (www.youtube.com/@LinusGroh), one of the developers.

» COMMUNITY AND SUPPORT

SerenityOS has an unusually active community by the standards of alternative OSes. It's refreshing to see contributors on a project like this who are actually making regular updates and posting videos about their progress.

The official website (<https://serenityos.org>) isn't exactly packed with traditional documentation at the moment, and it works as more of a hub

pointing to other resources. The GitHub page (<https://github.com/SerenityOS/serenity>) contains instructions for how to build and run SerenityOS. If you have questions about using the operating system, the best place to ask is the Discord server, which has a lot of activity from the friendly community.

YouTube is great for finding out more about SerenityOS and getting the latest

updates about what the developers are working on. It's also often possible to catch a live coding stream to watch a developer actually solving bugs and adding new features while explaining. Lead developer Andreas Kling regularly posts updates, in which he has edited in updates from other developers as well, on his YouTube channel (www.youtube.com/c/andreaskling).

QUICK TIP

There are a few games included with SerenityOS, and they mostly fall into the category of simple distractions and implementations of classic game types. There are some more substantial games, such as *Quake* and *Doom* engine ports, in the *Ports* directory in the source code tree.

```

anon@courage:~/Documents/3D Models
courage:~$ ls
Desktop  Documents  Downloads  Example_HTML.html
Games    README.md  Source     Tests
Videos
courage:~$ cd Documents/
courage:~/Documents$ ls
3D Models  Example_Presentation.presenter
emoji-serenity.txt  emoji.txt
pdf        eq
tips.txt      zip
courage:~/Documents$ cd 3D Models/
courage:~/Documents/3D Models$ ls
ladypall.obj  teapot.bmp  teapot.obj  tuba.obj
courage:~/Documents/3D Models$ ls /proc/
11 12 14 15 16 2 24 25 26 27 28 29
3 30 31 32 33 34 35 36 37 38 39 4
41 43 46 47 48 49 5 50 51 52 6 63
7 8 9 self
courage:~/Documents/3D Models$

```

Thankfully, the SerenityOS command-line interface leans more towards Unix than DOS. Most of the familiar commands are present.

experience seems to be the holy grail for alternative OSes, and the main criteria we take into consideration are browser speed and website compatibility. At the time of writing, *Ladybird*'s speed leaves something to be desired, and it's a roll of the dice whether a given site will work. However, as with all things connected to SerenityOS, the pace of development is encouraging. A number of complicated sites such as Google News are navigable, although the layout isn't quite right, and full Discord support is a current goal of the developers.

Ladybird development is an interesting point of entry to discover a few things about the SerenityOS development community. It's quite interesting to watch one of the developers working on the browser code on a YouTube livestream, for instance, testing different sites and fixing the browser in real time.

Creating a new browser from scratch might seem unconventional, but SerenityOS has a browser that is

probably just as good as, if not better than, the RISC OS, Haiku and AROS operating systems that rely on existing technology such as a standard layout engine. As *Ladybird* can run under Linux, we wonder if it could end up being a viable inclusion to those other systems?

The kernel

Thankfully, the SerenityOS kernel is based on modern principles and it fully supports preemptive multitasking and memory protection. Like Linux, it's a monolithic design, meaning that some of the system facilities such as drivers are located with the same memory space as the kernel for good performance. SerenityOS is a 64-bit operating system and the 32-bit support that was once present is now largely deprecated.

The kernel also supports multithreading. In other words, a program can launch its own processes. To see why this improves performance, consider two parts of an app in a case where one part is blocked as it waits for a reply from a server on the network. In OSes that don't support multithreading, the whole program has to stop until that task is completed. As well as improving overall performance, multithreading also provides a smoother experience, as an application stops and starts less often and is more responsive.

If you remember our examinations of RISC OS and AROS, we noted that the lack of threading support at the kernel level is a barrier to porting some apps.

SerenityOS doesn't yet feature support for multiple CPU cores. This lack doesn't block the porting of any software, but working SMP (Symmetric Multi-Processing) support can often greatly accelerate the operation of complex software and improve the experience. Experimental SMP support is already working and USB support is also under development.

An Arm CPU port is ongoing, making us wonder if we might one day see SerenityOS as a possible Raspberry Pi operating system, although that seems like something that's still on the horizon for the project.

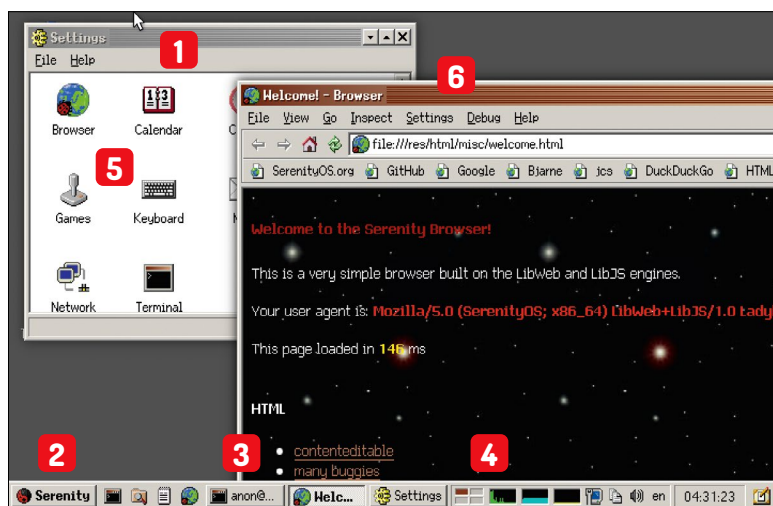
The competition

Comparing SerenityOS to other alternative desktops out there, our feeling is that it's heading in the right direction. Systems like the AmigaOS-derived AROS and RISC OS have huge legacies and a steep culture underpinning them. Apart from the nostalgia factor, it gives them a massive leg up when it comes to decades of accumulated knowledge of how they work and a large number of legacy apps that have been ported from the original systems. On the downside, both of those systems are held back by their legacy roots on the technical side, particularly in the area of the kernel.

If you want modern features and loads of software in every area, while staying away from the mainstream, a Unix-based operating system might be the most sensible option. Choices here include one of the desktop-orientated BSD systems and the Solaris-inspired OpenIndiana. The downside is that they might feel a bit too similar to Linux, which is a shame if you fancy tipping your toe into something really different.

HaikuOS is the most appropriate alternative operating system to compare SerenityOS to. Both have modern design features at their core and desktops that aspire to present the best of '90s style. In terms of being a complete system, HaikuOS is considerably

SAY HELLO TO SERENITYOS



1 Desktop look and feel
The overall aesthetic may remind you of popular desktops from the '90s.

2 Application launcher
At least sticking with a classic design means you can easily guess where the application launcher is.

3 Familiar elements
The standard task switcher is a tried-and-tested design, as are the quick launch icons.

4 System monitors
SerenityOS adopts modern accoutrements, such as handy monitors for the CPU, memory and network usage.

5 Configuration
The familiar working methods of this desktop even extend to system tools such as configuration utilities.

6 Applications
There are some impressive, homegrown apps, such as the *Ladybird* web browser.

ahead in features and maturity. It also has some really useful ports of large apps. The main shortcoming of HaikuOS is that it doesn't yet have a robust web browsing experience, and this seems to be one of the toughest nuts to crack for any alternative OS.

Daily driver capability

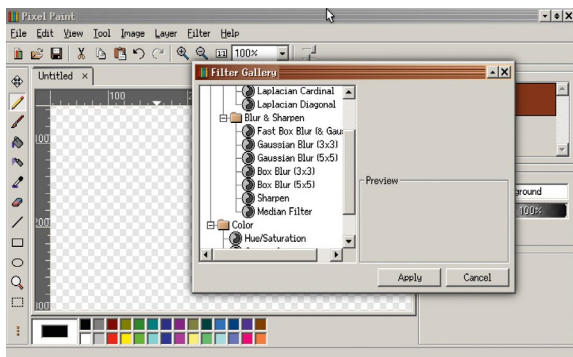
We wondered how close SerenityOS is to being a daily driver at this point, and to help us, we turned to the man himself, Andreas. "For me personally, I need good multi-core performance for programming projects, software and drivers for recording and editing videos, and a web browser that can handle all the things I do on the web daily. I'm working towards these things every day and it's going to take quite a while longer! Of course, someone with different requirements could start daily-driving the system much sooner. That's something I love about our project: we've got many people pulling towards their own personal goals for the system, and I'm just one of them."

That's an assessment that we are mostly in agreement with. However, we did encounter some stability problems from time to time, with occasional system crashes. We assume that these problems will be addressed as the system becomes more mature.

The rate that the community is developing the system is unusually fast for a small operating system project. For that reason, we can afford to give it the benefit of the doubt in a few areas. The development team has taken some unusual decisions in terms of developing so many applications and subsystems from scratch. But having said that, if it is a development process that is producing impressive results and rapid progress, it would be wrong to criticise it.

As it's a reasonably good target for ports, it should be unlikely to have any massive holes in terms of app types and utilities in the long term. That said, there are limits to what even a team as determined as SerenityOS's can manage, and we'd be excited to see more ports of GUI apps. Hopefully, that will develop in time as there is already a port of the Qt toolkit.

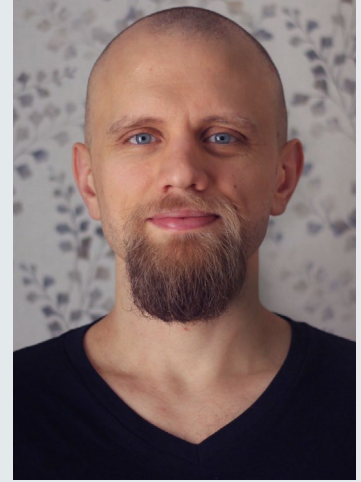
We'd judge the project to be in an alpha state at the moment, but it's intriguing and fun to experiment with. In its current state, it's a compelling enough system to get enthusiasts to take a look, and it must be nearly time for the community to create some downloadable ISOs so that more people can look it over. **LXF**



Pixel Paint looks like a clone of the famous Microsoft Paint, but it has some advanced features, such as layers and filters.

» TALKING TO THE DEVELOPER

Lead developer Andreas helped us fill in some technical details about how SerenityOS works, but we were interested to find out how the community came together and how the development of the OS has affected him personally.



Q. Were you surprised how popular your YouTube videos have turned out to be?

Definitely! I originally started making videos to help with some personal insecurities, and hoped that maybe someone would find them interesting. I never expected to find an audience in the tens of thousands!

Q. What have been some of the great moments during development in the SerenityOS project? We're talking about the emotional milestones. Pinging a server on the outside world? Booting to a desktop?

Those were both great milestones! Hearing sound for the first time was also cool. And compiling a C++ program! Later, porting classic games like *Doom* and *Diablo*. And the first time I did a Google search with our own browser. Most recently, taking the browser out of SerenityOS and making it run on other systems has been very fulfilling!

Q. The community around SerenityOS is impressive. What has been the formula for success there? Do you have any tips for other developers who want to build a supportive and productive community around their project?

I can't pretend to understand exactly why it's become so successful, but I do have three things in particular to offer as tips:

1. Create content! I've been putting out a steady stream of content about the project in the form of videos, streams, tweets, blogs and so on. Content is great at attracting new people, and makes the project feel alive.

2. Make a place to meet and collaborate! Communities are people, and people like to chat with each other. We used IRC up to 150 users, and then moved to Discord, where we now have over 7,000 users. Our spaces are highly focused on development, with designated areas for off-topic conversation.

3. Keep high standards! By holding ourselves to a high standard for code, version control, continuous integration and so on, we've scaled the project from one developer to hundreds while steadily increasing overall quality.

Q. When we looked at Haiku, we were impressed with the ports of applications like LibreOffice and Krita. If you could choose a couple of big-name apps to port to SerenityOS, what might they be?

To be honest, I'm much more interested in building our own applications instead of porting existing ones. If I had to choose something, it would be fun to have a digital audio workstation like *Reason* or *Ableton Live*, and also get *Steam* working so I could play *Dark Souls* on SerenityOS.

» **THERE ARE NO LXF ALTERNATIVES** Subscribe now at <http://bit.ly/LinuxFormat>

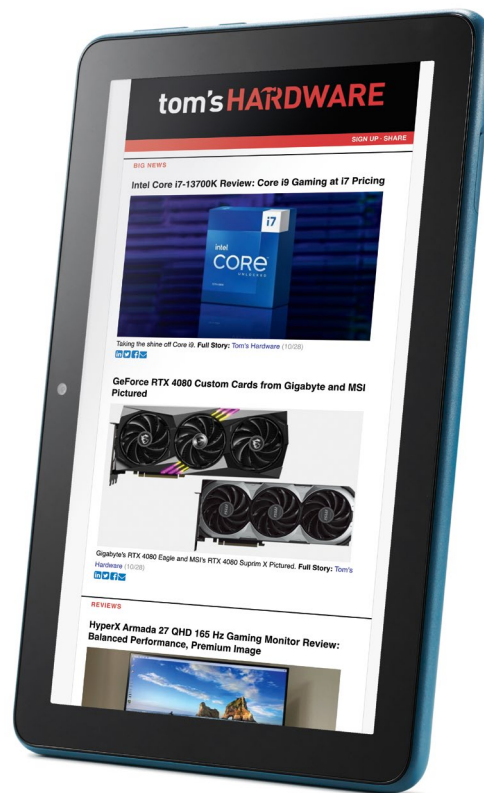
tom's **HARDWARE**

**GET ALL THE ESSENTIAL BREAKING
NEWS FOR THE TECH ENTHUSIAST!**

No matter if you're building a PC,
buying a laptop or learning about
robots, Tom's Hardware has all the
comprehensive knowledge you need.



**Scan & Subscribe
for free!**



The home of technology
techradar.com

HotPicks

Cryptomator » PowerShell » JOSM » Logseq » Easy Effects » Ear Tag » Gabut Download Manager » 3D-Puzzles » Total Chaos » Metadata Cleaner » Hidamari



Mayank Sharma

is good at finding brilliant open source software, but can't figure out what to do until F1 2023 (and binge-watching *Drive to Survive* doesn't count).

CLOUD DATA ENCRYPTOR

Cryptomator

Version: 1.6.15

Web: <https://cryptomator.org>

Most cloud data providers only encrypt data during transmission. Usually the ones who do encrypt the data, hold on to the decryption keys, leaving users little option but to trust the cloud providers to take adequate steps to keep the keys from being stolen, copied or misused.

If this arrangement has put you off using online data silos, *Cryptomator* encrypts your data before handing it over to the cloud silo. This ensures only you can decrypt your data, which remains undecipherable for anyone else, be it the cloud provider or a threat actor.

Cryptomator works with any cloud data storage service as long as it gives you the ability to synchronise with a local directory in your filesystem. This means it can work with virtually all popular services, including Dropbox, Google Drive, OneDrive, MEGA, pCloud, OwnCloud and Nextcloud.

Ubuntu users can install *Cryptomator* via a PPA, but your best bet is to use the official AppImage or Flatpak.

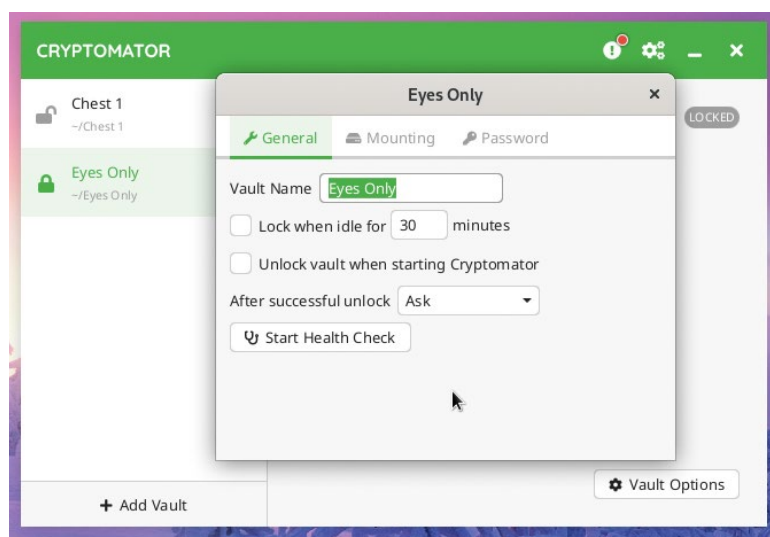
To install the app via Flathub, fire up a terminal and type `flatpak install flathub org.cryptomator`.

Cryptomator to install the app. You can then launch the app from the app launcher or with the `flatpak run org.cryptomator.Cryptomator` command.

The app has an intuitive interface. You first have to create an encrypted folder – or a vault, as the app refers to it – within your filesystem. This is password-protected, and everything inside it is encrypted using AES encryption with 256-bit key length.

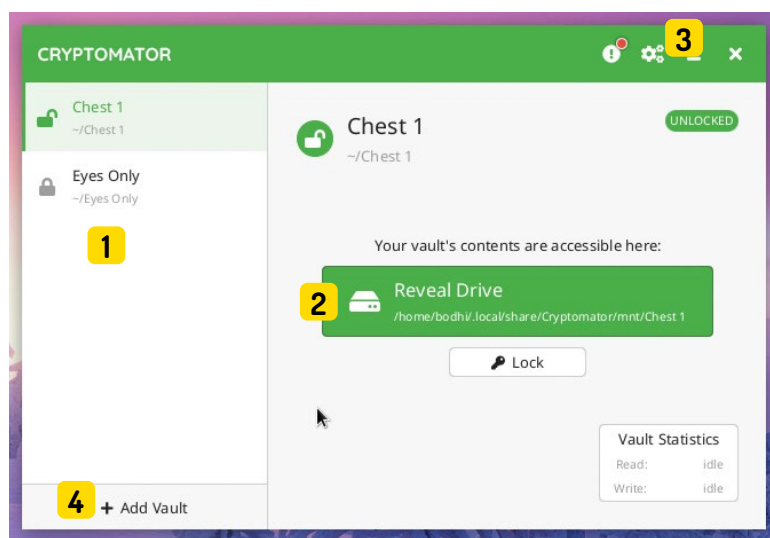
Now all you need to do is configure your cloud storage service to synchronise this encrypted folder with the cloud drive, and you're done.

Every time you store something on this drive, *Cryptomator* encrypts the data automatically, including the files names and file structure. The cloud storage service does the rest and makes sure your encrypted data is synced with the online silo. You can create multiple vaults – each one is protected by a password and can contain as many files and folders as you like.



Cryptomator is open source and has been audited by security researchers to ensure it doesn't contain any back doors.

LET'S EXPLORE CRYPTOMATOR...



1 Encrypted folders

All encrypted folders, whether locked or unlocked, are listed here. Right-click on a locked vault to reveal its options.

2 Unlocked drive

Once unlocked, you can use the Reveal Drive button to open the vault in the file manager, and also view its complete path in your filesystem.

3 App preferences

This button reveals the few settings for the app, such as the option to choose where to store the passwords, and make the app lock all open vaults automatically on exit.

4 Add encrypted folder

This button launches a brief wizard to help you create an encrypted folder in your filesystem.

CLI SHELL

PowerShell

Version: 7.3.0 Web: <https://github.com/PowerShell/PowerShell>

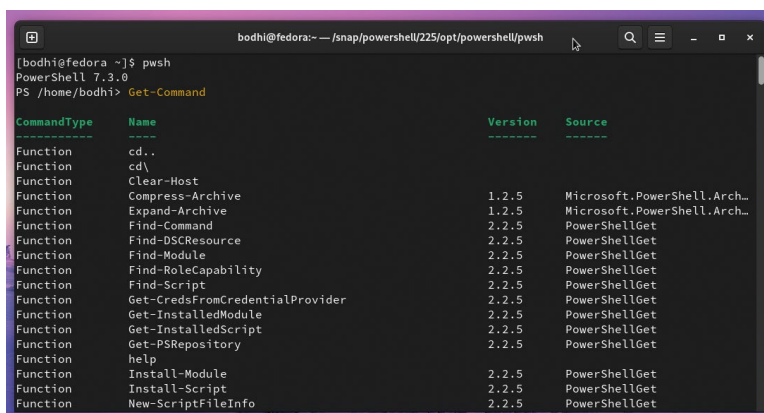
Free and open source software from Microsoft that's actually good is a rarity. One such gem is *PowerShell*, which interestingly supports all of the commonly known commands from *CMD* and Linux's command-line shell.

PowerShell is made up of a command-line shell, a scripting language and a configuration management framework. System admins use it for automating the management of the computers in their realm.

One of the differences that'll strike first-time users when using *PowerShell* is the semantic clarity.

As anyone who has worked with traditional POSIX shells like *BASH* knows, most commands are heavily abbreviated. For instance, commands such as **awk**, **ps** and **top** don't make sense to a new user as the names don't reveal the purpose of the command.

In contrast, commands in *PowerShell* are self-descriptive. They are called cmdlets and follow the verb-noun naming convention. For instance, the cmdlet **Get-Childitem | Where-Object Length -gt 50MB** lists all files greater than 50MB in the current



directory. That said, there are plenty of scenarios where standard Linux commands make more sense than *PowerShell* cmdlets. However, what's particularly appealing about *PowerShell* is that it lets you use commands and cmdlets together.

For instance, consider **Get-Process | Select-Object -Property Name, Path, Responding, Id | grep firefox**.

Even though **Get-Process** returns a number of properties for processes running on the computer, we pipe the output through **Select-Object** to narrow down the list, which is further narrowed to display only details about the running *Firefox* process using **grep**.

Binary packages for *PowerShell* are available for several distros, but you can install it from the Snap store with **sudo snap install powershell --classic**. Once installed, use **pwsh** to switch to *PowerShell*.

Use the "Get-Command" cmdlet to view a list of the available Linux PowerShell cmdlets.

OSM EDITOR

JOSM

Version: 22.10

Web: <https://josm.openstreetmap.de>

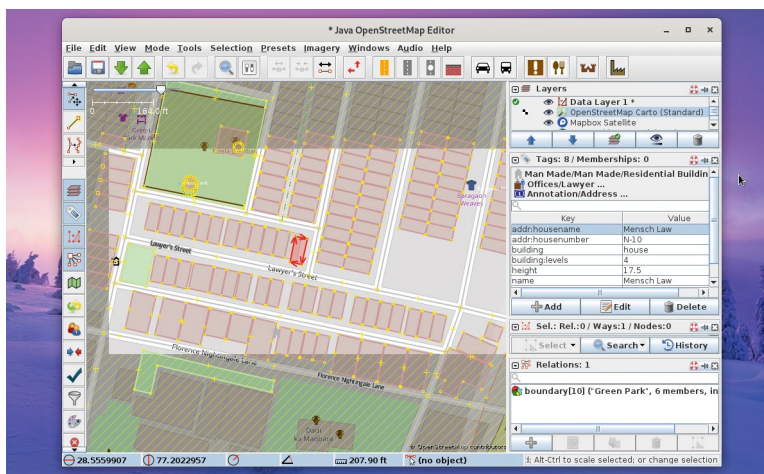
Although we're big fans of OpenStreetMap (OSM), its coverage isn't as extensive as the proprietary map app from Google. No worries; being a crowd-sourced project, it doesn't take much to become a cartographer and extend its scope.

JOSM, or the *Java OpenStreetMap Editor*, is a feature-rich, Java-based offline desktop map editor for OSM that can help you plot GPS traces.

JOSM is available in the repos of several distros, but you can use the official distro-agnostic Flatpak to make sure you have the latest version. Head to the terminal and type **flatpak install flathub org.openstreetmap.josm** to install the app. It anchors itself in the distro's Applications menu and can also be launched with **flatpak run org.openstreetmap.josm**.

You can upload your own GPS track data into *JOSM*, though it's best to grab the map data from the OSM database and enrich it by adding nodes, ways and tags.

The first step is to download some maps. You can practise using *JOSM* by downloading the map for your neighbourhood or an area you're familiar with or have



visited recently. To do so, click on the green download arrow and zoom into the map until you find the area you want to map. Select the area and click the Download button to fetch the existing data for that area from the main server. For most places on Earth there is also at least one image available. You can pick one from the Imagery menu at the top of the editor.

JOSM is an offline editor, which means everything you do won't be visible for anyone else until you upload it to the server. This makes it possible to experiment and repeatedly move, tag, add and delete elements without breaking anything. Once you've made the changes, click the green upload arrow to ferry the changes to OSM.

Using *JOSM* doesn't take much effort, but please read some getting-started documentation if you haven't edited OSM maps before.

KNOWLEDGE PLATFORM

Logseq

Version: 0.8.11

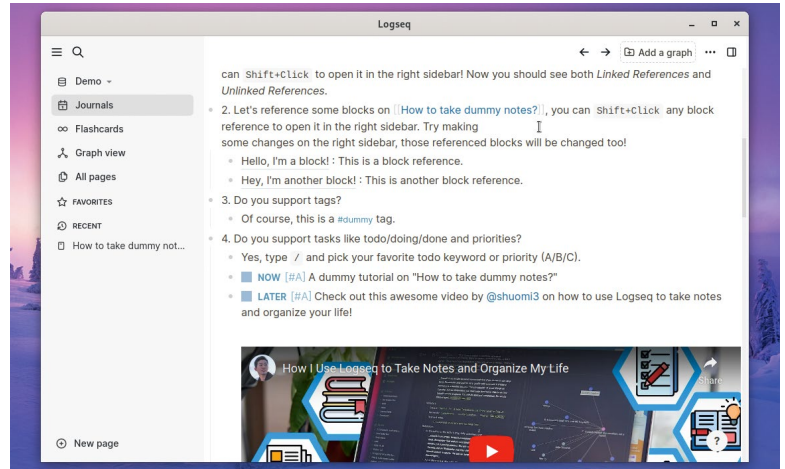
Web: <https://logseq.com>

In its most basic form, *Logseq* is a note-taking application. But there's got to be more to this software that separates it from the myriad of other apps for jotting down notes, because *Logseq* counts Google, Meta and MIT amongst its users.

You can use *Logseq* to jot down notes, and also add images and other documents such as PDFs. The app also enables you to organise notes using tags, which makes them easily searchable. *Logseq* is built with a privacy-first mindset, which means that by default all your notes are stored offline on your computer. That said, you do have the option to synchronise them using GitHub or iCloud.

For every release, *Logseq* puts out an official AppImage, which you can grab yourself and make executable by using the `chmod +x` command, or from the file manager.

At first glance, the app might appear as a regular note-taking app, but what makes it different is its ability to link and refer to other notes. We advise you to refer to its getting-started documentation to



familiarise yourself with its distinguishing features and how to use them.

Logseq claims the problem with popular note-taking apps is their inability to resurface information. Sure, old notes appear when you search for them, but with *Logseq*, you can link to old ones directly in the note itself, which makes it easier to find interlinked info. The project claims this helps you easily traverse your notes. It also has a view that shows all your notes as a graph, to help you visualise the arrangement of your notes.

The project claims its ability to combine the disparate notes, research, PDF markups and more can help you create a better and efficient system for storing and retrieving information.

Part of *Logseq*'s charm is that it supports markup languages including both Markdown and Emacs' Org Mode.

AUDIO MANIPULATION

Easy Effects

Version: 7.0.0 Web: <https://github.com/wmm/easyeffects>

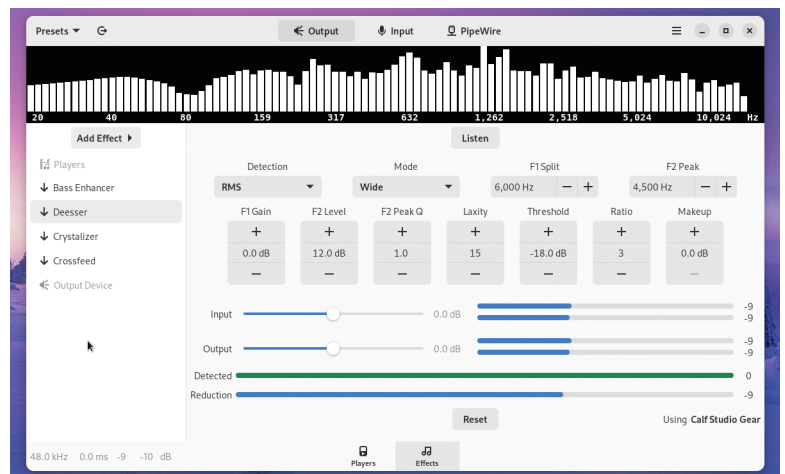
Audiophiles take note. To get the most oomph from your expensive audio gear, you can use the *Easy Effects* audio manipulation tool. As its name suggests, the tool helps you apply audio effects to apps managed by PipeWire, with little effort.

It includes dozens of effects, such as an equaliser, exciter, limiter, compressor, reverb tool and lots more.

To complement these it also features a built-in spectrum analyser. With its latest version, you can use the app to not just include multiple effects to a stream, but also change their order as per your audio pleasure.

The tool is included in the repos of several distros but to make sure you have the latest version, just use the Flatpak with `flatpak install flathub com.github.wmm.easyeffects`. Once it's installed, launch it from the distro's Applications menu or with `flatpak run com.github.wmm.easyeffects`.

Audio enthusiasts won't have any trouble navigating and using the app. Still, there's a comprehensive manual included, with details about the parameters for each of the included effects, which is pretty useful if



you aren't an audiophile. Because *Easy Effects* uses the default PipeWire sound server, its developers claim it works with most, if not all, apps on a modern distro.

One convenient feature we particularly like is that *Easy Effects* remembers the configuration from the last session, and restores it automatically when launched. Of course, you can also save your custom arrangements as profiles that you can call upon whenever required in the future.

Besides manipulating sound output, *Easy Effects* can also apply effects to an input device, such as a microphone. This isn't just useful when recording audio, but can also help enhance conversations during online meetings and presentations.

The app lists all music-oozing apps in the main window, from where you can enable or disable them as per your needs.

TAG EDITOR

Ear Tag

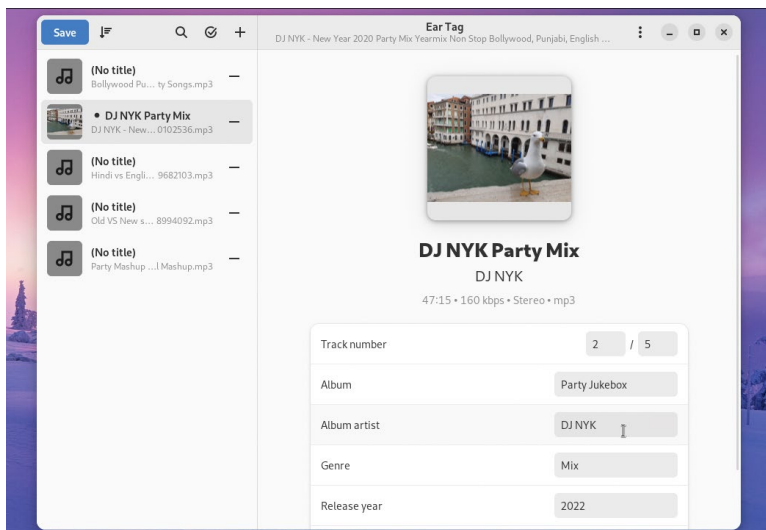
Version: 0.2.1 Web: <https://github.com/knuxify/eartag>

Ear Tag is a small and simple music tag editor, which unlike many of its peers, doesn't try to manage your entire library. Its developers claim that the application is suitable for editing a handful of files or even a single file, and is particularly useful for people who don't have a very large library of music files.

Ear Tag is available in Arch's AUR and a couple of other distros. Your best bet, however, is to install it via the official Flatpak package with `flatpak install flathub app.drey.EarTag`. Once it's installed, you can fire up it from the distro's application launcher or from the CLI with `flatpak run app.drey.EarTag`.

Being a GTK4 application, Ear Tag is fairly attractive and looks at home in modern Gnome-based distros such as Ubuntu and Fedora, and like other native apps, can also follow the system's colour scheme to automatically switch between light and dark themes.

To use the app, either drag and drop your music file into the app or point it to the file by navigating your filesystem from within the app. You can open multiple



files, but if you have a rather large collection, it's best to look at something like Picard.

If you open multiple files, make sure the one you want to edit is selected on the left. This displays the file's existing tag data, including some irreplaceable information such as length of track, type and encoding.

You can now change the cover image of the file, as well as use the text fields to edit other metadata information, such as the title, name of artist, the album, track number, genre and more. Once you've made the changes, click the Save button to make sure the updated tag information is saved to the file.

Thanks to LibAdwaita and the app's compact design, it can be used on any screen size, including mobile Linux devices.

DOWNLOAD MANAGER

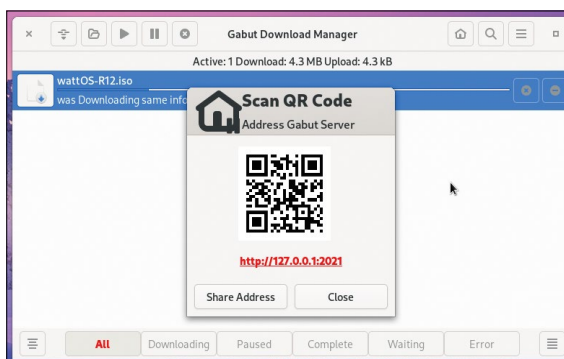
Gabut Download Manager

Version: 1.9.9 Web: <https://github.com/gabutakut/gabutdm>

Gabut Download Manager is a feature-rich all-in-one download manager that can grab files using direct URLs, as well as from Metallinks, Magnetlinks and Torrents. The icing on the cake is its built-in web service, which not only remote-controls the manager but also share files over the local network.

Gabut isn't available in distro repos but you can grab it from the Snap store or Flathub. To install the Flatpak version, head to the terminal and type `flatpak install flathub com.github.gabutakut.gabutdm`. When done, you can launch it from the distro's application menu or with `flatpak run com.github.gabutakut.gabutdm`.

The app actively monitors your clipboard and automatically catches hold of any downloadable URLs



and opens the download dialog box. Its interface resembles that of any download manager, and gives all the usual information about ongoing downloads.

Gabut can also help you pass authentication details if asked by the remote server, as with FTP downloads.

Its distinguishing feature is its local web service. When you click the home icon in the interface, it pops up a window with a QR code along with an IP address. You can scan the QR code from your mobile phone or head to the displayed IP address from the web browser on any other computer on the local network.

The default user and password for remote access are `gabutdm` and `123456`. From the web interface, you can manage the downloads remotely, transfer files from the device you are accessing the remote interface on to the host that's running Gabut, as well as transfer files from the computer with Gabut to the device on which you are viewing the web interface.

You can change default authentication information and enable folder sharing from under the app's Settings window.

CUBE PUZZLES

3D-Puzzles

Version: N/A **Web:** <https://github.com/volatileflorovium/3D-Puzzles>

As its name suggests *3D-Puzzles* is a simulator of 3D combination puzzle games. In addition to the classic Rubik's Cube, the game also features the Dologic, and Dologic 12 puzzles.

To get started, download its official AppImage from the project's website:

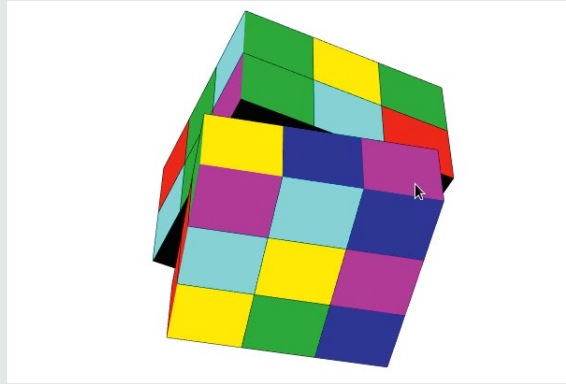
```
$ wget https://github.com/volatileflorovium/3D-Puzzles/raw/main/3D-Puzzles-x86_64.AppImage
```

After grabbing the file, you have to make it executable, which can be done in either of two ways. Firstly, you can use the graphical user interface by right-clicking on the file and selecting Properties from the context menu, then you need to head to the Permissions tab and toggle the Allow Executing File As Program checkbox.

Or you can fire up the terminal and switch to the folder where you've downloaded the AppImage:

```
$ chmod +x 3D-Puzzles-x86_64.AppImage
```

The game has a simple interface. The main window displays all the puzzles. Click the one you want to play. Irrespective of the puzzle you select, you're presented



The game is functional but doesn't offer conveniences like the ability to save multiple games or to select a difficulty level.

with an option to either select a new game or load the previous session.

The current session is saved automatically when you exit the puzzle. But if you select to start a new game, the previous session is lost. If there is no previous session, the puzzle is scrambled on launch.

To move the pieces, click near any edge of a piece and drag the mouse parallel to the edge in the desired direction, while holding down the button. This takes some getting used to, especially in Dologic 12, which has a large number of movable pieces. You can also use the arrow keys to change puzzle view.

Although this is the game's first release, if you enjoy solving 3D puzzles, it's a must-have. As the author acknowledges, the intention of the game isn't to produce stellar graphics, but rather to be functional, and it works as advertised.

FIRST-PERSON SHOOTER

Total Chaos

Version: 1.2.1 **Web:** www.moddb.com/mods/total-chaos

There's no shortage of *Doom 2* mods, but *Total Chaos* is such a brilliant remaster that you wouldn't have guessed its *Doom 2* roots if we hadn't spilled the beans.

The first-person shooter has six levels and does a stellar job of transforming *Doom 2* into a horror-themed survival bloodfest.

It relies on *GZDoom*, which is a 3D-accelerated *Doom* port that makes best use of the current hardware to add support for things such as dynamic lighting, sloped floors and other things that modernise the game's visuals and make the gameplay more engaging overall.

You can grab the game from Flathub with `flatpak install flathub com.moddb.TotalChaos`. When it's installed, you can run it from the distro's application menu or by using the `flatpak run com.moddb.TotalChaos` command.

Total Chaos is set on the fictitious Fort Oasis island in the midst of a wind-battered sea. The island was once home to a buzzing coal mining community, which



has now disappeared without a trace, leaving the facility abandoned.

The game launches with you waking up on the island after wrecking your ship. As you get your bearings, a voice on the walkie-talkie urges you to find them. To make your way around, you must fight your way through all kinds of horrifying creatures, each with their own unique attack patterns.

You fight through apartments, utility tunnels and other buildings before making your way to the city centre. On the way, you unravel the mystery of what happened through letters scattered around, along with a large battery of weapons. Sure, there are guns, but mostly you have to defend yourself with tyre irons, pickaxes and such. A normal day at work!

Total Chaos is really well done, and does a stellar job of building tension, with its eerie soundscape and the usual tricks popular with the genre.

PRIVACY TOOL

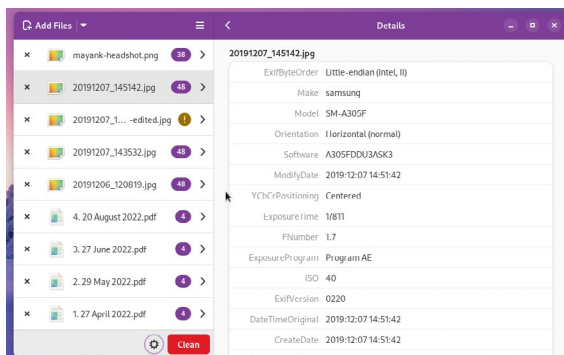
Metadata Cleaner

Version: 2.2.7 Web: <https://metadatacleaner.romainvigier.fr>

As long-time readers can probably tell, we're suckers for privacy here at **LXF Towers**. One of the sloppiest ways to leak privacy information is through metadata in files posted in public forums. Metadata is sensitive information that, with the right context, can reveal a lot about you.

Cameras record data about when and where a picture was taken and which camera was used. Similarly, many office apps automatically add author and other details to all documents. With *Metadata Cleaner*, you can easily view the metadata embedded in a file and remove it without much effort.

The app recommends installing it via Flathub. Pull up a terminal and type `flatpak install flathub fr.romainvigier.MetadataCleaner` to install it. When it's done, you can launch it from the distro's Applications menu or from the terminal with `flatpak run fr.romainvigier.MetadataCleaner`.



The app has an approachable interface that inexperienced users will find welcoming.

You can use the Add Files button in the app to select a file you wish to sanitise or bring up the pull-down menu adjacent to the button to point the app to a folder and allow it to load all files in there.

After it's done analysing the files you've added, you can click on them to view the metadata the app has found within each file. To clean all the files in the window, press the Clean button.

The developers warn that in its bid to be thorough with the purging, the removal process could modify the files in unexpected ways. However, you can change this default behaviour by switching the app to the Light mode, which isn't as harsh and doesn't make any destructive changes to the files.

The app has a detailed manual that does a good job of explaining the various options and points out why removing metadata isn't a foolproof mechanism for ensuring that files don't leak personal data.

VIDEO WALLPAPERS

Hidamari

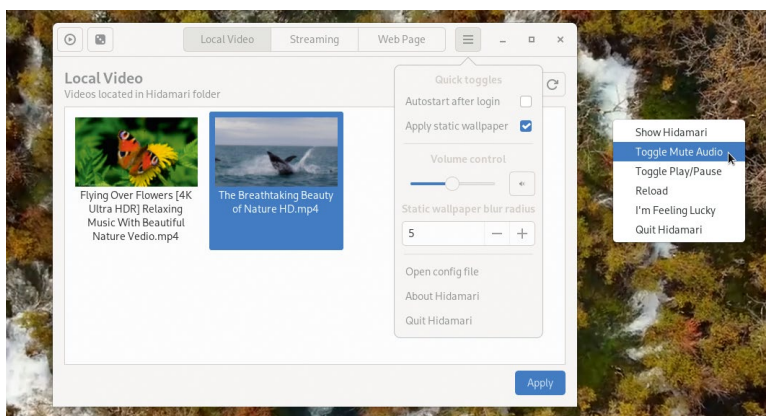
Version: 3.2 Web: <https://github.com/jeffshee/hidamari>

Static wallpapers are so passé, especially on modern machines with dedicated graphics hardware. With *Hidamari*, you can run videos as wallpaper, without taxing your computer.

The Python app praises itself for being minimal, and yet offers features such as GPU video decoding. Best of all, it's fully compatible with Wayland and supports multi-monitor setups as well.

Hidamari is available in the Arch's AUR, but the developers suggest installing it via the distro-agnostic Flatpak. Bring up a terminal and type `flatpak install flathub io.github.jeffshee.Hidamari` to install the app. It then nests itself in the distro's Applications menu and can also be launched from the terminal with `flatpak run io.github.jeffshee.Hidamari`.

Besides setting local videos as wallpaper, the app also enables you to use videos from YouTube and other video-sharing sites, as well as set web pages as wallpaper. These options are exposed through three separate tabs in the app. To play any of your videos as wallpaper, drop them inside the `~/Videos/Hidamari`

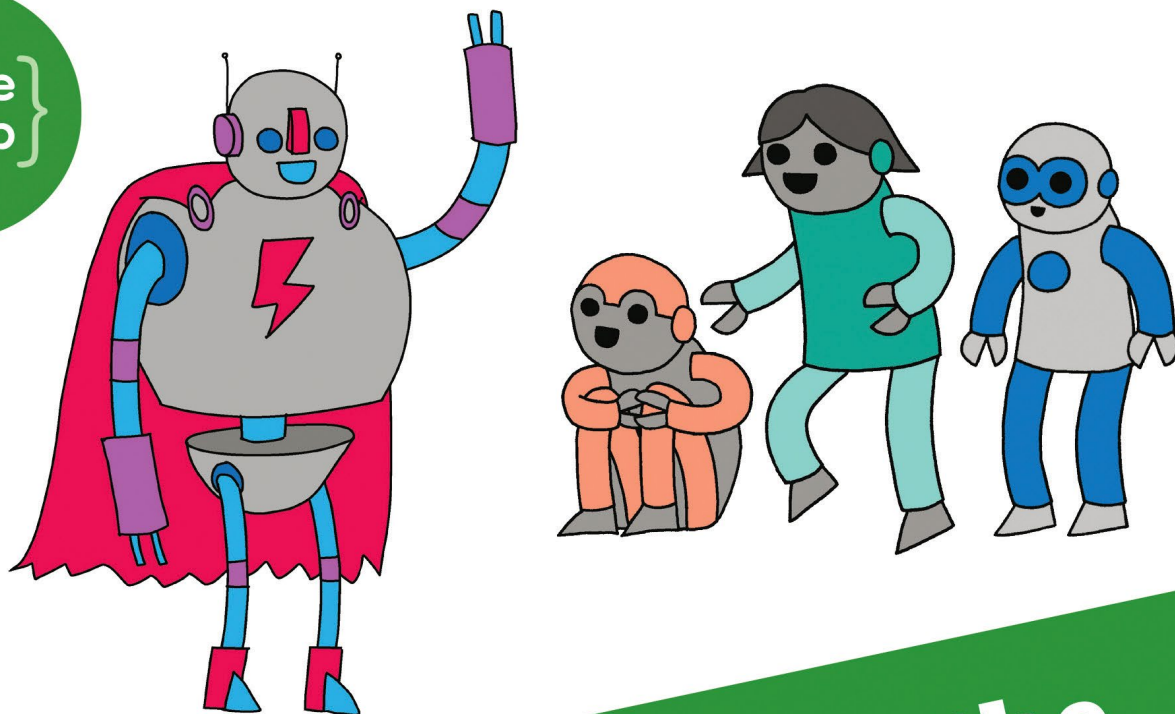
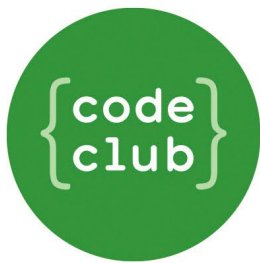


folder. Any files in there show up under the Local Video tab. Select the video you want to use as the wallpaper and click the Apply button.

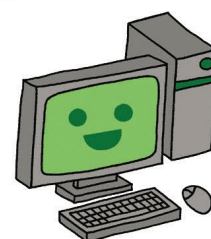
To use a video from the internet as wallpaper, switch to the Streaming tab and paste the URL of the video in the textbox. Similarly, head to the Web Page tab to point to a website or an HTML file on your computer to use it as your wallpaper.

Once the wallpaper is set, you can right-click on the desktop to bring up relevant options, such as to mute and pause playback. Other options, such as the ability to control the volume of the video, are all accessible from the hamburger menu in the app. Here you'll also find toggles to autostart the software and start the video wallpaper as soon as you log into your desktop. **LXF**

The app is smart and automatically pauses playback when a window is maximised or an app is running in full-screen mode.



**Can you help inspire the
next generation of coders?**



Code Club is a nationwide network of volunteer-led after school clubs for children aged 9-11.

We're always looking for people with coding skills to volunteer to run a club at their local primary school, library or community centre for an hour a week.

You can team up with colleagues, a teacher will be there to support you and we provide all the materials you'll need to help get children excited about digital making.

There are loads of ways to get involved!
So to find out more, join us at **www.codeclub.org.uk**

EBPF

Credit: www.ebpf.io

Monitor your TCP/IP networks with eBPF

Naturally suspicious, **Mihalis Tsoukalos** keeps an eye on all his network traffic using eBPF – and he'll show you how, too.



OUR EXPERT

Mihalis Tsoukalos is a systems engineer and a technical writer. He is the author of *Go Systems Programming and Mastering Go*, 3rd edition.

Sure to send a shiver down the nerdiest of geek's spines, the idea of monitoring TCP/IP packets and connections can be confusing at best. Just scan this article and the output of the eBPF tools, and you realise you need to be familiar with TCP/IP and its cryptic options in order to understand the provided information. Before diving into the TCP/IP tools, we'll begin by explaining how to trace eBPF related system calls with the help of the *Strace* utility.

Let's use *Strace* to trace eBPF-related system calls along with their parameters. For example, we can find out the eBPF system calls executed with *Opensnoop*:

```
$ sudo strace -e bpf /usr/share/bcc/tools/opensnoop
bpf(BPF_MAP_CREATE, {map_type=BPF_MAP_TYPE_PERF_EVENT_ARRAY, key_size=4, value_size=4, max_entries=8, map_flags=0, inner_map_fd=0, map_name="events", map_ifindex=0, btf_fd=0, btf_key_type_id=0, btf_value_type_id=0, btf_vmlinux_value_type_id=0, map_extra=0}, 72) = 4
bpf(BPF_MAP_UPDATE_ELEM, {map_fd=4, key=0x7f86f7370610, value=0x7f86f7370690, flags=BPF_ANY}, 144) = 0
bpf(BPF_RAW_TRACEPOINT_OPEN, {raw_tracepoint={name=NULL, prog_fd=7}}, 144) = 8
```

This is a small part of the generated output but you get the idea and the insights of the system calls related to eBPF. Using filtering tools such as *Grep* can help you locate what you are looking for because *Strace* produces lots of output.

What we see here is three versions of the BPF system call. The first one using the `BPF_MAP_CREATE` command, the second one using `BPF_MAP_UPDATE`

```
Tracing connect ... Hit Ctrl-C to end
PID COMM IP SADDR DADDR DPORT
962 KCupsConnect 6 ::1 ::1 631
962 KCupsConnect 4 127.0.0.1 127.0.0.1 631
1295 Chrome_Child 4 192.168.1.7 40.114.211.99 443
1295 Chrome_Child 4 192.168.1.7 13.94.251.244 443
1295 Chrome_Child 6 2a02:587:2f03:2300:c014:9781:d4b8:3455 26
1b:2a9 443
4752 ssh 4 192.168.1.7 195.251.38.180 22
4756 ssh 4 192.168.1.7 109.74.193.253 22
4760 Socket Threa 6 2a02:587:2f03:2300:c014:9781:d4b8:3455 26
4760 Socket Threa 4 192.168.1.7 52.85.158.55 443
4760 Socket Threa 6 2a02:587:2f03:2300:c014:9781:d4b8:3455 26
0:93a1 443
4760 Socket Threa 4 192.168.1.7 34.107.221.82 80
4760 Socket Threa 6 2a02:587:2f03:2300:c014:9781:d4b8:3455 26
4760 Socket Threa 4 192.168.1.7 52.85.158.55 443
4760 Socket Threa 6 2a02:587:2f03:2300:c014:9781:d4b8:3455 2a
4760 Socket Threa 4 192.168.1.7 34.120.208.123 443
4760 Socket Threa 4 192.168.1.7 34.120.208.123 443
4760 Socket Threa 4 192.168.1.7 34.212.166.60 443
4760 Socket Threa 6 2a02:587:2f03:2300:c014:9781:d4b8:3455 2a
```

Here you can see the output from the *TCPconnect* utility, which traces TCP connect calls.

ELEM and the last one using `BPF_RAW_TRACEPOINT_OPEN`. In **LXF297**, we discussed eBPF maps and how each map has a file descriptor associated with it – this can be seen in action here: the first call creates a map and returns a file descriptor with a value of 4, whereas the second statement uses 4 as its `map_fd` parameter in order to update the desired map.

Traditional TCP/IP tools

Before deciding to use eBPF utilities, consider using one of the existing Unix ones. The list of such tools includes but is not limited to the following:

- **Netstat**: Shows information about the network status of a Unix machine. It's a powerful tool that can work on the Socket, TCP, UDP, IP and Ethernet levels. Its main drawback is that it only works on the local machine but doesn't usually need root to operate.
- **SS (Socket Statistics)**: A modern version of *Netstat*. When used without any command-line arguments, SS prints all TCP, UDP and socket connections.
- **TCPdump**: Allows you to capture network data using the capabilities of the *libpcap* library. However, it cannot analyse the captured network data.

PID	COMM	LADDR	LPORT	RADDR	RPORT	TX_KB	RX_KB	MS
5867	ssh	192.168.1.7	47414	195.251.38.180	22	12	6	1715.04
1080	skypeforli	192.168.1.7	54268	13.107.3.128	443	3	0	73.51
480	NetworkMan	192.168.1.7	33678	95.216.195.133	80	0	0	176.15
480	NetworkMan	2a02:587:2f03:2300:c014:9781:d4b8:3455	49308	2a01:4f9:c010:2636::1	80	0	0	174.94
5888	sshd	192.168.1.7	22	192.168.1.3	56842	3	3	7912.90
5921	Socket Thr	192.168.1.7	46476	34.120.208.123	443	0	3	311.32
5921	Socket Thr	2a02:587:2f03:2300:c014:9781:d4b8:3455	60444	2606:4700:20::ac43:485a	443	3	27	276.24
5921	Socket Thr	192.168.1.7	35766	93.184.220.29	80	0	0	5218.58
5921	Socket Thr	192.168.1.7	35758	93.184.220.29	80	0	0	5220.42
5921	Socket Thr	2a02:587:2f03:2300:c014:9781:d4b8:3455	45464	2a00:1450:4001:828::2008	443	1	46	578.50
5921	Socket Thr	2a02:587:2f03:2300:c014:9781:d4b8:3455	37570	2a00:1450:4001:82f::200e	443			

● **WireShark:** As well as *Tshark*, the command-line version of *WireShark*, it can help you capture but most importantly analyse TCP/IP traffic. *WireShark* is more or less the standard tool for looking into TCP/IP traffic at the network packet level.

● **/proc/net** This system folder has lots of useful information, provided that you read up about it. The following output shows the kind of information you should expect to get from **/proc/net/protocols**:

```
$ sudo cat /proc/net/protocols
```

protocol	size	sockets	memory	press	maxhdr	slab	module
UDIPv6	1344	3	1	NI	0	yes	kernel
TCPv6	2392	3	1	no	320	yes	kernel
UDP	1152	1	1	NI	0	yes	kernel
TCP	2232	5	1	no	320	yes	kernel

In here we see information about the UDIPv6, TCPv6, UDP and TCP protocols (we've removed some output for brevity).

TCP utilities...

The *TCPconnect* tool traces TCP connections that are initiated with the **connect** system call. The generated output looks as follows:

PID	COMM	IP	SADDR	DADDR	DPORT
5481	ssh	4	192.168.1.7	195.251.38.18	22
5484	ssh	4	192.168.1.7	109.74.193.253	22
4760	Socket Threa	4	192.168.1.7	35.163.114.24	443

Sometimes, there is a delay in the serving of **connect** system calls. Should you wish to learn more about that latency, you can look at the *TCPconnlat* utility, which generates the following kind of output:

```
$ sudo /usr/share/bcc/tools/tcpconnlat
```

PID	COMM	IP	SADDR	DADDR	DPORT	LAT(ms)
962	KCupsConnect	4	127.0.0.1	127.0.0.1	631	0.10
4752	ssh	4	192.168.1.7	195.251.38.180	22	27.47
4756	ssh	4	192.168.1.7	109.74.193.253	22	66.75
4760	Socket Threa	4	192.168.1.7	54.187.120.38	443	252.38
4760	Socket Threa	4	192.168.1.7	54.187.120.38	443	240.15
4760	Socket Threa	4	192.168.1.7	54.187.120.38	443	248.26

Looking at the first line of output, you might think that the remaining connections are pretty slow. However, this is not the case because the first connection is on the local machine. Additionally, the output of *TCPconnlat* contains almost all of the information found in the default output of *TCPconnect*.

Output of "tcptop -C" – like "top" for TCP connections. The -C flag doesn't clear the screen when printing new information.

However, *TCPconnect* supports a richer set of command-line options than *TCPconnlat* offers.

The *TCPstates* tool traces TCP session state changes and durations. Its output looks like:

```
SKADDR C-PID C-COMM LADDR LPORT
RADDR RPORT OLDSTATE
-> NEWSTATE MS
ffff98abb689180 5630 ssh 192.168.1.7 0
195.251.38.180 22 CLOSE
-> SYN_SENT 0.000
ffff98abb689180 347 irq/165-iw 192.168.1.7 40744
195.251.38.180 22 FIN_WAIT1
-> FIN_WAIT2 32.431
```

A particularly interesting feature of *TCPstates* is that it presents both the old and the new state of a TCP connection. Additionally, *TCPstates* supports many command line-options that can modify the information that is contained in the output. So, **tcpstates -4** traces IPv4 connections only, whereas **tcpstates -6** traces IPv6 connections only.

Please be sure to run **\$ sudo /usr/share/bcc/tools/tcpstates --help** or look at the source code of the tool to learn more about the rest of the supported command line options.

TCPlife and TCPtop

The *TCPlife* tool traces the lifespan of TCP sessions and creates a summary for each completed TCP connection. This tool is handy for viewing the list of completed TCP connections. The output from *TCPlife* looks similar to the following:

PID	COMM	LADDR	LPORT	RADDR	RPORT	TX_KB
-----	------	-------	-------	-------	-------	-------

QUICK TIP

You can find many UDP eBPF tools along with their implementation in the *BPF Performance Tools* book by Brendan Gregg. Generally speaking, this is the best book to learn more about eBPF. However, be prepared for lots of reading and experimenting.

» TCP/IP BOOT CAMP

TCP/IP is a family of protocols that help the internet to operate. The name comes from its two most well-known protocols: TCP and IP. TCP stands for Transmission Control Protocol, TCP software transmits data between machines using segments, which are also called TCP packets. The main characteristic of TCP is that it is a reliable protocol, which means that it makes sure that a packet was delivered without needing any extra code from the programmer. If there is no proof of packet delivery, TCP sends that

particular packet again – this continues until there is proof of delivery.

When a TCP connection is made between two machines, a full duplex virtual circuit, similar to a phone call, is created between those two machines. They constantly communicate to make sure data is sent and received correctly. If the connection fails for some reason, the two machines try to find the problem and report to the relevant application. The TCP header of each packet includes the source port and destination port fields. These two fields, plus the source

and destination IP addresses, are combined to uniquely identify every TCP connection. All these details are handled by TCP/IP as long as you provide the required details without any extra effort.

IP stands for Internet Protocol and it is not a reliable protocol by nature. IP encapsulates the data that travels over a TCP/IP network because it is responsible for delivering packets from the source host to the destination host according to the IP addresses. IP must find an addressing method for sending a packet to its destination effectively.



usecs	count	distribution
0 -> 1	: 0	
2 -> 3	: 0	
4 -> 7	: 0	
8 -> 15	: 0	
16 -> 31	: 0	
32 -> 63	: 0	
64 -> 127	: 0	
128 -> 255	: 0	
256 -> 511	: 0	
512 -> 1023	: 0	
1024 -> 2047	: 0	
2048 -> 4095	: 0	
4096 -> 8191	: 0	
8192 -> 16383	: 0	
16384 -> 32767	: 10	****
32768 -> 65535	: 95	*****
65536 -> 131071	: 29	*****
131072 -> 262143	: 6	**

Here you can see the output generated by the `TCPrtr` utility that calculates TCP RTT.

RX_KB	MS
5867	ssh 192.168.1.7 47414 195.251.38.180 22 12
6	1715.04
1080	skypeforli 192.168.1.7 54268 13.107.3.128 443 3
0	73.51

QUICK TIP

The status of a TCP connection from the client side can be one of SYN-Sent, ESTABLISHED, FIN-WAIT-1, FIN-WAIT-2 or TIME_WAIT. The status of a TCP connection from the server side can be one of LISTEN, SYN-RCVD, CLOSE-WAIT, LAST-ACK, CLOSED or CLOSING.

Because *TCPlife* generates lots of output, especially when dealing with HTTP traffic, you might need to filter the generated output on your own to be able to use it effectively.

The *TCPtr* tool works like *Top* but for TCP connections. It shows a summary of TCP send/rcv throughput by host. The produced output of *TCPtr* looks as follows:

PID	COMM	LADDR	RADDR	RX_KB	TX_KB
9032	b'sshd'	192.168.1.7:22	192.168.1.3:63540	0	0

If you execute *TCPtr* using the `-C` command-line option, you can see all the presented information – this enables you to process the collected data more easily. Type `$ sudo /usr/share/bcc/tools/tcptop --help` to see all the supported command-line options.

TCPaccept.bt and TCPretrans

As its name suggests, *TCPaccept.bt* traces successful `accept()` system calls. The kind of output you should expect from *TCPaccept.bt* is as follows:

TIME	PID	COMM	RADDR	RPORT	LADDR	LPORT	BL
23:33:17	524	sshd	127.0.0.1	50662	127.0.0.1	22	0/128
23:33:38	524	sshd	192.168.1.3	56962	192.168.1.7	22	0/128

The reason for using *TCPaccept.bt* instead of the BCC version (*TCPaccept*) is because *TCPaccept* does not print any output, which looks like a bug that might be fixed in future versions.

The *TCPretrans* tool traces TCP retransmits. The generated output looks as follows:

TIME	PID	IP	LADDR:LPORT	T>
------	-----	----	-------------	----

RADDR:RPORT	STATE
15:32:40 0 4 192.168.1.7:39450 R> 93.184.220.29:80	SYN_SENT
15:32:46 0 4 192.168.1.7:50136 R> 194.63.247.211:443	FIN_WAIT1

The last column offers valuable information because it presents the state of the TCP connection. A high number of TCP retransmits is an indication of a busy network, a network/hardware device problem, or both. You might need to stress-test your local network by downloading multiple large files from the internet or other computers that belong to the same network in order to find out how your network behaves under heavy load. Additionally, you can use a tool such as *AB*, which is the Apache HTTP server benchmarking tool, to make multiple HTTP requests and see how your network behaves.

Using *TCPretrans* with the `-c` flag is going to present a summary of retransmits as soon as you stop its execution by pressing `Ctrl+C`.

The TCPtracer utility

The *TCPtracer* tool traces TCP connections, which means that you should expect to get lots of output from it. Therefore, this should not be the first tool to execute when trying to resolve TCP issues. The kind of output generated by *TCPtracer* is as follows:

Tracing TCP established connections. `Ctrl-C` to end.

T	PID	COMM	IP	SADDR	DADDR	SPORT	DPORT
C	3169	pacman	6	[2a02:587:2f03:2300:c014:9781:d4b8:3455]	[2606:4700:20::6819:5e05]	39498	443
C	3169	pacman	6	[2a02:587:2f03:2300:c014:9781:d4b8:3455]	[2a0c:59c0:6::100]	54742	443
X	3169	pacman	6	[2a02:587:2f03:2300:c014:9781:d4b8:3455]	[2606:4700:20::6819:5e05]	39498	443
X	3169	pacman	6	[2a02:587:2f03:2300:c014:9781:d4b8:3455]	[2a0c:59c0:6::100]	54742	443

The previous output shows the IPv6 connections (the `x::x::x::x::x` addresses) created by *Pacman* (the Arch Linux package manager) when running an update.

Netqtop and Gethostlatency

The *Netqtop* utility traces the kernel functions performing packet transmit and packet receive on the data link layer, which is the reason that it needs a network interface to operate. On a Lenovo ThinkPad, for instance, the name of the Wi-Fi interface is `wlp0s20f3`. So, we are running *Netqtop* using the `--name wlp0s20f3` command line argument. The generated output looks as follows:

`$ sudo /usr/share/bcc/tools/netqtop --name wlp0s20f3`
Thu Jun 23 22:28:49 2022

TX
QueueID avg_size [0,64] [64,512] [512,2K] [2K,16K] [16K,64K]
0 338.0 0 2 1 0 0
Total 400.0 0 1 1 0 0
RX
QueueID avg_size [0,64] [64,512] [512,2K] [2K,16K] [16K,64K]
0 52.0 2 0 0 0 0
Total 52.0 2 0 0 0 0

This is pretty low level information that can be handy when you suspect that there might be an issue

Output in the form of a histogram generated by a BCC one-liner.

size	count	distribution
0 -> 1	: 0	
2 -> 3	: 0	
4 -> 7	: 0	
8 -> 15	: 0	
16 -> 31	: 32	*****
32 -> 63	: 94	*****
64 -> 127	: 4	*
128 -> 255	: 3	*
256 -> 511	: 1	
512 -> 1023	: 1	
1024 -> 2047	: 7	**


```

~ (ssh)
+ ~ sudo bpfttrace -e 'k:tcp_sendmsg { @send_bytes = hist(arg2); }'
Attaching 1 probe...
^C

@send_bytes:
[16, 32] 19 |#####|
[32, 64] 36 |#####|
[64, 128] 34 |#####|
[128, 256] 11 |#####|
[256, 512] 12 |#####|
[512, 1K] 18 |#####|
[1K, 2K] 4 |####|
[2K, 4K] 2 |##|
[4K, 8K] 1 |#|

+ ~ sudo bpfttrace -e 'k:udp_recvmmsg { @send_bytes = hist(arg2); }'
Attaching 1 probe...
^C

@send_bytes:
[1K, 2K] 32 |#####|
[2K, 4K] 4 |####|
[4K, 8K] 0 |
[8K, 16K] 0 |

```

The output from two BPFtrace one-liners, the first working with “tcp_sendmsg” and the second working with “udp_recvmmsg”.

with one of your network interfaces. TX is for data sending, whereas RX is for data receiving.

In the previous output, it looks as though the network interface is working without any issues. However, the most accurate criterion is to try the tool on a regular basis to get a better understanding of what 'normal' and 'working' look like for a given Linux machine in order to better recognise abnormal situations, which is a great tactic in general.

The *Gethostlatency* utility traces the latency of DNS calls by generating output in the following format:

TIME	PID	COMM	LATms	HOST
18:09:00	1295	ThreadPoolForeg	27.06	static.asm. skype.com
18:09:00	1295	ThreadPoolForeg	22.14	static-asm. secure.skypeassets.com
18:11:39	12715	ssh	0.71	uop
18:11:53	12752	pacman	647.72	cdn.repo. archlinuxcn.org

What does the output tell us? First, it tells us that Skype keeps asking DNS questions even though it is not being used on a call. Then, it tells us that the **uop** hostname is resolved extremely fast, which makes perfect sense because it is hardcoded in **/etc/hosts**. Last, it informs us that it takes 647.72ms to resolve the address of **cdn.repo.archlinuxcn.org**, which is pretty slow. We can do many things based on the previous information. First, close Skype, second, put **cdn.repo.archlinuxcn.org** in **/etc/hosts**, third, find a different way to make its resolve time much faster because this host is used for updating Arch Linux.

As mentioned before, DNS can create difficult-to-find networking issues, so having a tool for calculating the latency of DNS host resolution can be pretty handy.

BCC one-liners

Here are some handy BCC one-liners. The first counts all UDPv6 functions and generates this kind of output:

```
$ sudo /usr/share/bcc/tools/funccount 'udp6*'
Tracing 13 functions for "b'udp6*'" ... Hit Ctrl-C to end.
FUNC                                COUNT
b'udp6_proc_exit'                   5
b'udp6_proc_init'                   13
```

You can use `udp_*` to work with UDPv4, `tcp_*` to

» APPROACHING NETWORKING ISSUES

All the tools you need are here – usually more than we need – but how do you approach a networking issue rationally? Such a strategy might include the following steps:

1. Begin by trying to understand your issues using some of the traditional Unix tools. Usually, *Netstat*, *Tcpdump* or *WireShark* are all you need to get a better idea of what's wrong.
2. Get statistics about open TCP/IP connections, packets and connection rates. Use tools such as *Netstat* and *SS*.
3. Check the network interfaces of the system. Especially the utilisation percent of each network interface.
4. Look for DNS issues – DNS can create nasty and difficult-to-find TCP/IP errors. Use tools such as *Host*, *Dig* and *Nslookup*.
5. Calculate network latency. If needed, create synthetic traffic.
6. Start using eBPF tools to go deeper into your search. This includes tracing new and existing TCP connections.
7. Keep looking – please look for DNS issues one more time!
8. Try *WireShark* or *Tshark* to look into actual TCP/IP traffic.

Generally speaking, networking issues can be from fairly simple to extremely difficult to resolve, so please be patient.

work with TCPv4 and `tcp6*` to work with TCPv6.

The second one-liner counts TCP retransmits:

```
$ sudo /usr/share/bcc/tools/funccount 't:tcp:tcp_
retransmit_*
```

The final BCC one-liner calculates the UDP sent bytes using a histogram:

```
$ sudo /usr/share/bcc/tools/argdist -H 'p::udp.  
sendmsg(void *sk,void *msg,int size)::int:size'
```

The screenshot (*bottom-left*) shows sample output from the previous one-liner, which can give you a very good idea of the UDP traffic on your system.

Bpftrace one liners

Finally we're going to present some handy *Bpfttrace* one-liners. The first one counts TCP sends and receives per application and process ID, and generates the following kind of output:

```
$ sudo bpftrace -e 'k:tcp_send[func,pid,comm] = count(); }
```

Attaching 2 probes...

```
@[tcp sendmsg,1365,skypeforlinux]: 1
```

```
@[tcp_recvmsg,9613,sshd]: 1
```

```
@[tcp_sendmsg,9613,sshd]: 1
```

```
@[tcp_recvmmsg,1365,skypeforlinux]: 3
```

Once again, *Skype for Linux* generates traffic in order to check for updates on chats. The second one-liner generates a histogram of TCP send bytes:

```
$ sudo bpftrace -e 'k:tcp_sendmsg { @send_bytes = hist(arg2); }'
```

The final *Bpftrace* one-liner generates a histogram of UDP receive bytes:

```
$ sudo bpftrace -e 'k:udp_recvmmsg { @send_bytes = hist(arq2); }'
```

You have learned a great deal about using eBPF for TCP/IP connections and your horizons have been expanded a lot. However, you are going to forget them if you do not run and experiment with the presented commands on your own. LXF

QUICK TIP

WireShark is the ultimate tool for looking into TCP/IP traffic and analysing network traffic of almost any kind. You can learn more about WireShark at www.wireshark.org and by looking at its documentation site at <https://www.wireshark.org/docs/>.

» **IMPROVE YOUR CODE SKILLS** Subscribe now at <http://bit.ly/LinuxFormat>

PYTHON/DJANGO

Credit: www.djangoproject.com

Finish your cataloguing app

Part Three!

Don't miss
last issue,
order it on
page 68!

With a tear in his eye, **Matt Holder** finishes up our cataloguing database app by adding search and export functionality.



OUR
EXPERT

Matt Holder has been a fan of the open source methodology for over two decades and uses Linux and other tools where possible. In his spare time, Matt enjoys listening to music and reading.

In this series of articles, we've been working on a cataloguing application, where magazines, newspapers and corresponding articles can be added, stored and eventually searched. In parts one and two, we introduced Django, described some basic database theory, created our database models, explored the built-in shell, added some URLs and created some views and HTML pages to view output from the software. We also explored the admin interface, which allows data to be added and deleted.

In this final article, we're exploring the forms functionality, which allows for HTML forms to be added to our web pages to provide data entry, and then we're creating a page to enable us to search the database and return articles, authors and publications.

A completed project can be downloaded from GitHub (see the *QuickTip*, below-left) and contains the functionality discussed in the three articles and more.

First, let's create a search page. This will consist of a URL to link against the Python view, a Python view to provide the functionality to search the database, and an HTML page to display results. This will also use the forms functionality to provide the text and other fields before taking the input and using it in search queries, and then the data will be output and displayed in the browser, along with links to open the original record.

Before entering the URL into the browser, make sure you enter `python ../manage.py runserver` into the terminal, then the built-in web server is invoked.

When using forms in Django, there are some key terms we need to know about. A form can be bound or unbound. An unbound form doesn't contain any data, while a bound one does contain data, which can be processed by a view. The flow of using a form is as follows: firstly a **ModelForm** object is created in **forms.py**, which contains any fields that we want to gather data on. In this case, there are entries for the search term, a choice field to determine which database table or model to search, and a button that allows for a fuzzy search. Next, an instance of this **ModelForm** needs to be created in our search view, which is in **views.py**. The **ModelForm** object is then passed into our context object when loading the HTML template. Within the

Publication Database

Search the Database

- [Search the Database](#)

View data in the database

- [Articles](#)
- [Tags](#)
- [Authors](#)
- [Publications and Publishers](#)
- [Editions](#)

Add Data to the database

- [Add Tag to the Database](#)

Export Data from the database

- [Export Tags to CSV](#)

By the end of this project, we will have reproduced this functionality.

HTML template there is a template item called `{{form}}` which is included and this ensures that the fields within the **ModelForm** object are drawn on the page. Opening and closing the `<form>` tags are done within the HTML template as well as including the URL to post data to as well as adding the Submit button.

First we need to add a URL to the **urls.py** document. The code below needs to the **urlpatterns** list:

```
urlpatterns = [
    ...
    path('search/', views.search, name='search'),
]
```

Next, we need to create **forms.py** and add some code, which enables us to create our search form:

```
from django import forms
class SearchForm(forms.Form):
    typeChoices = [
        ('TAGS', 'Tags'),
        ('ARTICLES', 'Articles'),
        ('AUTHORS', 'Authors'),
        ('PUBLICATIONS', 'Publications'),
        ('PUBLISHERS', 'Publishers'),
        ('ALL', 'All')]

```

QUICK TIP

More feature-packed source code for this project can be downloaded from <https://github.com/mattmole/LXF-Django-Project>.

```
searchTerm = forms.CharField(label='Search Term',
max_length=100,required=False)
searchModel = forms.ChoiceField(label='Area to
Search', choices=typeChoices)
fuzzySearch = forms.BooleanField(label='Fuzzy
Search',required=False)
```

The format is very similar to that of a model. This is because models and form objects can work hand in hand and deal with similar sorts of information. First of all, we import the necessary libraries and then define a class, which inherits from the `forms.Form` library. We then create a list of tuples to be used for a multiple-choice field. In each tuple, we have a human-readable value and a value that is used in the POST data that gets passed to the model. Next we define an object for each form item, which will be shown on the screen. Each object is then linked to a HTML form object.

In the next code, we can see the HTML included in the `search.html` file, which needs to be added to the `Catalogue/templates/Catalogue` directory.

```
<!DOCTYPE html>
<html lang="">
<head>
<meta charset="utf-8">
<title>List of Tags</title>
</head>
<body>
<h1>Search the Database</h1>
<p>All results will be shown for a given model or
all models by using the fuzzy search option and leaving
the searchTerm blank or using the * character</p>
<form action="{% url 'search' %}" method="post">
{% csrf_token %}
{{ form.as_p }}
<input type="submit" value="Submit">
</form>
</body>
</html>
```

What's happening here is that the Jinja-templated HTML is being used to build a page containing the usual `<head>` information and then in the `<body>` we add a `<form>` tag, add a CSRF token to invoke the built-in Cross Site Request Forgery support, render the form from the view code (see below) and then add a Submit button, before we close all the necessary tags.

The next stage is that we need to write the Python code to link the URL to the program logic, which is used to both render the empty search form and then take the submitted data and perform a database search. Add the function below to the `views.py` file.

```
from .forms import SearchForm
from itertools import chain
def search(request):
# if this is a POST request we need to process the
form data
if request.method == 'POST':
# create a form instance and populate it with data
from the request:
form = SearchForm(request.POST)
# check whether it's valid:
if form.is_valid():
searchTerm = form.cleaned_data['searchTerm']
searchModel = form.cleaned_
data['searchModel']
fuzzySearch = form.cleaned_data['fuzzySearch']
```

Search the Database

All results will be shown for a given model or all models by using the fuzzy search option and leaving the `searchTerm` blank or using the `*` character

Search Term:

Area to Search: Tags ▼

Fuzzy Search: ☐

```
searchResults = None
if fuzzySearch == True:
if searchModel == 'TAGS':
if searchTerm == "" or searchTerm == "*":
tagResults = Tag.objects.all()
else:
tagResults = Tag.objects.filter(tagName__
contains = searchTerm)
searchResults = tagResults
elif searchModel == 'ALL':
if searchTerm == "" or searchTerm == "*":
searchResults = chain(Tag.objects.all(),
Article.objects.all(),
Author.objects.all(),
Publication.objects.all(),
Publisher.objects.all()
)
else:
searchResults = chain(Tag.objects.
filter(tagName__contains = searchTerm),
```

A search page used to provide criteria for database searching.

QUICK TIP

Custom validations can be included when writing models: <https://docs.djangoproject.com/en/4.0/releases/4.0/#functional-unique-constraints>.

» CUSTOM VALIDATIONS

When creating the model for tags, as explained in article one of the series, the only field that is required to be unique is that of the `id` value. Interestingly, when defining the model, the `id` field does not need to be defined, because this is automatically included by the database ORM. This means that multiple tags of the same name can be added to the database, which isn't optimal. When creating the models, fields can have the `unique=True` option set. While this is a step closer to what we want, it would not stop two tags being registered with the same name, but different casing of the letters. This is where custom validations become useful. See the Quick Tip (above) for further information, but to ensure that only one tag, representing Linux, can be added to the database, a constraint like the one below needs to be added to the model.

```
class Tag(models.Model):
tagName = models.CharField("Tag", max_length = 20)
tagDescriptor = models.CharField("Description", max_length = 200,
blank=True)

class Meta:
constraints = [
UniqueConstraint(
Lower('tagName'),
name='tagNameUnique',
),
]
```



QUICK TIP

The reportlab library can be used to generate PDFs for output: <https://docs.djangoproject.com/en/4.1/howto/outputting-pdf/>.

```
Article.objects.filter(articleName__
contains = searchTerm),
Author.objects.filter(authorName__
contains = searchTerm),
Publication.objects.
filter(publicationName__contains = searchTerm),
Publisher.objects.filter(publisherName__
contains = searchTerm)
)
else:
    if searchModel == 'TAGS':
        tagResults = Tag.objects.filter(tagName =
searchTerm)
        searchResults = tagResults
    elif searchModel == 'ALL':
        searchResults = chain(Tag.objects.
filter(tagName = searchTerm),
Article.objects.filter(articleName =
searchTerm),
Author.objects.filter(authorName =
searchTerm),
Publication.objects.filter(publicationName =
searchTerm),
Publisher.objects.filter(publisherName =
searchTerm)
)
    template = loader.get_template('Catalogue/
searchResults.html')
    context = {
        'searchResults': searchResults
    }
    return HttpResponse(template.render(context,
request))
# if a GET (or any other method) we'll create a blank
form
else:
    form = SearchForm()
    return render(request, 'Catalogue/search.html',
{'form': form})
```

The first thing that we have to do, within this function, is to determine whether it is being called using the **POST** or **GET** method. We test this then add the necessary function logic to search the database based on the conditions shown in the data passed into the function. This data is being described in the **searchTerm**, **searchModel** and **fuzzySearch** variables.

Next we use the **searchModel** variable to determine which database queries to run. Printed here is a cut-down function, due to space constraints. When selecting the Tags option in the search form, a database query runs to return **Tag** objects, when Articles is selected, **Article** objects are returned, and

so on. In this part of the function's logic, we also determine whether the fuzzy search option has been selected. The difference here is that instead of using **Tag.objects.filter(tagName = searchTerm)** to return objects that match exactly, we use **Tag.objects.filter(tagName__contains = searchTerm)** to return objects that contain the search term. Also, when using the model type of **ALL**, we need to be able to search all models for relevant objects. To accomplish this, we test for a **modelType** of **ALL** and then perform a database search in all models and use the chain function to join them all together, so that all results are stored in one object, which can be passed to the search results page.

Finally, when returning the search results, we construct a context dictionary, which contains the search results, and then use the **HttpResponse** function to render the **searchResults.html** template, while also giving it access to the search results data. Should the function be called from a **GET** method, we simply render the **search.html** template because there has been no data provided to the function and we are rendering the form, ready for the user to fill in.

Now that the logic has been described to allow the database to be searched, we need to create the template that will render the database results. You need to create a file called **searchResults.html** in the **Catalogue/templates/Catalogue** directory and add the following contents:

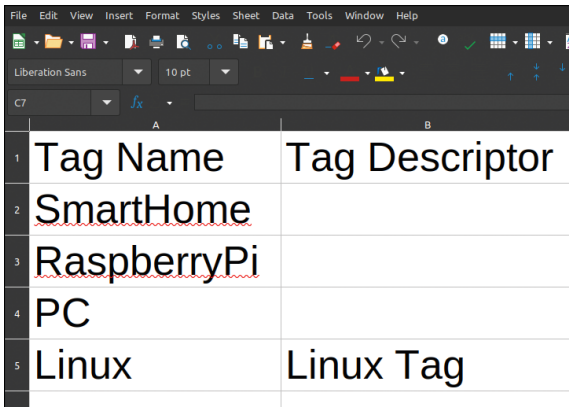
```
<!DOCTYPE html>
<html lang="">
<head>
<meta charset="utf-8">
<title>List of Search Results</title>
</head>
<body>
<h1>List of Search Results</h1>
{% if searchResults %}
<table>
<thead>
<tr>
<th>Result Type</th>
<th>Result Name</th>
</tr>
</thead>
{% for result in searchResults %}
<tr>
<td>{{result.modelType}}</td>
{% if result.modelType == "Tag" %}
<td><a href="{% url 'tag' result.id %}" >{{ result.
tagName }}</a></td>
{% elif result.modelType == "Publication" %}
<td>{{ result.publicationName }}</td>
{% elif result.modelType == "Publisher" %}
<td>{{ result.publisherName }}</td>
{% elif result.modelType == "Article" %}
<td>{{ result.articleName }}</td>
{% elif result.modelType == "Author" %}
<td>{{ result.authorName }}</td>
{% endif %}
</tr>
{% endfor %}
</table>
{% else %}
<p>No results are available.</p>
{% endif %}
```

Searching the database for all tags provides a table of results.

List of Search Results

Result Type Result Name

Tag	SmartHome
Tag	RaspberryPi
Tag	Linux
Tag	PC



Tag Name	Tag Descriptor
SmartHome	
RaspberryPi	
PC	
Linux	Linux Tag

Data can be exported from the database in many different formats, with this example showing CSV.

```
</body>
</html>
```

As in previous template files, we first of all add the usual boilerplate HTML to define the language type and `<head>` information. Within the `<body>` tags, we first of all test if data is available. If it isn't, a message is rendered. If search results are available, we add them to a table within a new row. As the search results can be from a range of different database objects, the required URL to view information about the result needs to be known. This is handled using the `result.modelType` variable from within the template. This data is provided by adding a function to each model in the `models.py` file, where the function returns the name of the model/class. This simple function, when added to each model, accesses internal information about the class, so that the result type can be returned.

```
def modelType(self):
    return self.__class__.__name__
```

When the search button has been selected, a page, like the one in shown (*below-left*), is displayed. Test this by opening a web browser and visiting <http://localhost:8000/catalogue/search>.

In this final part of the article, we are creating the logic required to be able to export data from the database into a CSV file. Even though there are hundreds of file formats that can be used to export data, all the way from XML to JSON, a CSV file can still be incredibly useful and can be used in many different applications. The first thing that needs to be done is for another item to be added to the `urlpatterns` list within the `urls.py` file, within the **Catalogue** directory.

```
urlpatterns = [
    ...
    path('exportTags/', views.exportTagsToCsv,
         name='exportTagsToCsv')
]
```

You can see an example of the CSV export from the database (*above*). While the code detailed below is specifically for tags, it could easily be generalised to allow a choice of model to export from and then the export routine processes accordingly.

Within the `views.py` file, add the `exportTagsToCsv` function, save the file and this then provides a link between the URL we registered and this logic.

» TESTING FOR RELIABILITY

While not the most exciting thing in the world, testing is crucial to ensure that our programming projects work as they are intended. While the example we have been working on for the last few months is relatively simple, getting into the habit of testing is a really good thing. So, do as your author says and not as he does! This is where unit tests are really important. Unit tests are a way of creating automated tests to ensure that a given input to a function returns the correct result. A good example of this would be to ensure that it is not possible to create two tags with the same name. The first attempt to create the new tag should be successful and the second time should return a failure. Another example of a unit test is that when adding a publication with a type that is not in our list of choices, it should return a failure. You can learn more about unit tests at the following two web pages: <https://docs.djangoproject.com/en/4.1/topics/testing/overview/> and <https://docs.djangoproject.com/en/4.1/intro/tutorial05/>.

To invoke this code, in the browser, visit <http://localhost:8000/catalogue/exportTags>.

```
import csv
def exportTagsToCsv(request):
    tagList = Tag.objects.order_by('-tagName')
    response = HttpResponse(
        content_type='text/csv',
        headers={'Content-Disposition': 'attachment;
        filename="exportTags.csv"'},
    )
    writer = csv.writer(response)
    writer.writerow(['Tag Name', 'Tag Descriptor'])
    for tag in tagList:
        writer.writerow([tag.tagName, tag.tagDescriptor])
    return response
```

In this code entry, we are first of all exporting all **Tag** objects from the database and sorting it in reverse alphabetical order. We then create a **response** object, where we set the **content_type** and **headers**, so that the browser knows to expect a CSV file, rather than HTML content. Next we create a **csv.writer** object, which is linked to the **response** object, which we return at the end of the function. The next step is to write a row to the **writer** object, which contains the column headings. Next, we use a for loop to output the tag's name and tag's description for each item in the database. Finally, we return the **response** object. When the above URL is visited, a CSV file is downloaded.

Available on GitHub (see *Quick Tips*, page 94), there is a second method that allows data to be exported as CSV. This uses JavaScript to read the values of the results table and generates a CSV file, which can be downloaded. This second method is functional, but could run into issues, should the table be so large that the JavaScript uses a large amount of memory.

The aim with this series of articles has been to introduce some useful concepts, such as database theory, ORMs and the link between our models, views and controllers. We hope that you have enjoyed the series and can now envisage how to use the concepts learned in your own projects. **LXF**

QUICK TIP

Adding CSS to style the site is easy: <https://docs.djangoproject.com/en/4.1/intro/tutorial06/>.

» **IMPROVE YOUR CODE SKILLS** Subscribe now at <http://bit.ly/LinuxFormat>

NEXT MONTH

FREE YOUR PHONES!

Linux has the power to help revive old phones, keep them secure and running the latest apps.



LXF299
will be on sale
Tuesday
7th February
2023

Get packing

A new generation of packaging formats is upon us. Love them or loathe them, we take them for a spin.

Algorithmic art

From fractals to AI, learn how computer pixels are transforming what we think of as art and how you can give it a go.

Inside Fedora Silver Blue

Fedora is redefining the desktop distro, providing a more secure immutable filesystem, OSTree and Flatpak installs.

Linux next-gen gaming

Is the Steam Deck really that good and what can it do to replace your desktop? We reveal all and have some fun!

LINUX

The #1 open source mag **FORMAT**

Future Publishing Limited,
Quay House, The Ambury, Bath, BA1 1UA
Email contact@linuxformat.com

EDITORIAL

Editor-in-chief Neil Mohr
Art editor Fraser McDermott
Production editor Katharine Davies
Group editor-in-chief Graham Barlow
Group art director Jo Gulliver
Editorial contributors
Mike Bedford, Denise Bertacchi, Jonathan Bolding,
Neil Bothwick, Anj Bryant, Nate Drake, Karsten
Günther, Christian Guyton, Matthew Holder,
Christopher Lloyd, Jon Masters, Nick Peers,
Aaron Peters, Les Pounder, Michael Reed,
David Rutland, Mayank Sharma, Shashank Sharma,
Mats Tage Axelsson, Mihalis Tsoukalos
Cover illustration Magictorch.com
Ubuntu is a trademark of Canonical Limited. We are not endorsed
by or affiliated with Canonical Limited or the Ubuntu project.
Raspberry Pi is a trademark of the Raspberry Pi Foundation.
Tux credit: Larry Ewing (lewing@isc.tamu.edu).
Content production Adequate Media Limited

ADVERTISING

Commercial sales director Clare Dove
clare.dove@futurenet.com
Senior advertising manager Lara Jaggon
lara.jaggon@futurenet.com
Head of commercial – technology Dave Randall
dave.randall@futurenet.com
Account director Andrew Tilbury
andrew.tilbury@futurenet.com

INTERNATIONAL LICENSING

Head of print licensing Rachel Shaw
Linux Format is available for licensing and syndication.
To find our more contact us at licensing@futurenet.com
or view our content at www.futurecontenthub.com.

NEW SUBSCRIPTIONS & BACK ISSUES

Web www.magazinesdirect.com
UK 0330 333 1113 World +44 (0) 330 333 1113

EXISTING SUBSCRIPTIONS

Web www.mymagazine.co.uk
UK 0330 333 4333 World +44 (0) 330 333 4333
Subscription delays: Disruption remains within UK
and international delivery networks. Please allow up
to seven days before contacting us about a late
delivery to help@magazinesdirect.com

CIRCULATION

Head of newstrade Tim Mathers

PRODUCTION AND DISTRIBUTION
Head of production UK & US Mark Constance
Production project manager Clare Scott
Senior ad production manager Jo Crosby
Digital editions controller Jason Hudson

THE MANAGEMENT

Editorial director technology group Paul Newman
Content director Joe Cox
Head of art and design Rodney Dive
Design director Brett Lewis
Commercial finance director Dan Jotcham
Printed by Wyndeham Peterborough, Storey's Bar
Road, Peterborough, Cambridgeshire, PE15 5YS
Distributed by Marketforce, 5 Churchill Place,
Canary Wharf, London, E14 5HU
www.marketforce.co.uk Tel: 0203 787 9001

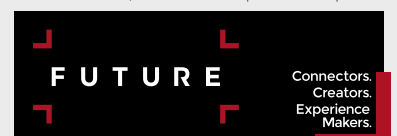
Linux® is the registered trademark of Linus Torvalds in the US and other countries.
GNU/Linux is abbreviated to Linux throughout the publication. Where applicable,
code printed in this magazine is licensed under the GNU GPL v2 or later. See
www.gnu.org/copyleft/gpl.html. All copyrights and trademarks are recognised
and respected.

Disclaimer All contents © 2023 Future Publishing Limited or published under
licence. All rights reserved. No part of this magazine may be used, stored,
transmitted or reproduced in any way without the prior written permission of the
publisher. Future Publishing Limited (company number 2008585) is registered in
England and Wales. Registered office: Quay House, The Ambury, Bath BA1 1UA. All
information contained in this publication is for information only and is, as far as we
are aware, correct at the time of going to press. Future Publishing Limited cannot
accept any responsibility for errors or inaccuracies in such information. You are
advised to contact manufacturers and retailers directly with regard to the price of
products or services referred to in this publication. Apps and websites mentioned in
this publication are not under our control. We are not responsible for their contents
or any other changes or updates to them. This magazine is fully independent and
not affiliated in any way with the companies mentioned herein.
If you submit material to us, you warrant that you own the material and/or have
the necessary rights/permissions to supply the material and you automatically
grant Future Publishing Limited and its licensees a licence to publish your
submission in whole or in part in any/all issues and/or editions of publications, in any
format published worldwide and on associated websites, social media channels
and associated products. Any material you submit is sent at your own risk and,
although every care is taken, neither Future Publishing nor its employees, agents,
subcontractors or licensees shall be liable for loss or damage. We assume all
unsolicited material is for publication unless otherwise stated and reserve the right
to edit, amend, adapt all submissions. All contents in this magazine are used at your
own risk. We accept no liability for any loss of data or damage to your systems,
peripherals or software through the use of any guide.

We are committed to only using magazine paper derived from
responsibly managed, certified forestry and chlorine-free
manufacture. The paper in this magazine was sourced and
produced from sustainable managed forests, conforming to
strict environmental and socioeconomic standards.

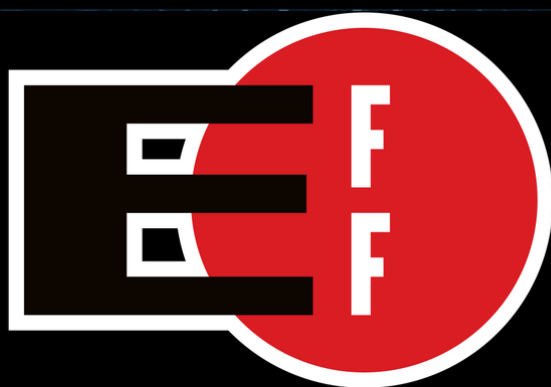


Future is an award-winning international media group and
leading digital business. We reach more than 57 million
international consumers a month and create world-class
content and advertising solutions for passionate
consumers online, on tablet & smartphone and in print.



Future plc is a public
company quoted on the
London Stock Exchange
(symbol: FUTR)
www.futureplc.com
Chief Executive Zillah Byng-Thorne
Non-Executive Chairman Richard Huntingford
Chief Financial and Strategy Officer Penny Ladkin-Brand
Tel +44 (0)1225 442 244





The Electronic Frontier Foundation is the leading nonprofit organization defending civil liberties in the digital world. Founded in 1990, EFF champions user privacy, free expression, and innovation through impact litigation, policy analysis, grassroots activism, and technology development. We work to ensure that rights and freedoms are enhanced and protected as our use of technology grows.

EFF.ORG

ELECTRONIC FRONTIER FOUNDATION

Protecting Rights and Promoting Freedom on the Electronic Frontier



**THE
BRAIN
TUMOUR
CHARITY**

A CURE CAN'T WAIT

**BRAIN TUMOURS
MOVE FAST.
WITH YOUR HELP,
WE CAN TOO!**

We're working to create a future where brain tumours are curable.
We urgently need your help to accelerate research.

Text DEFEAT5 to 70507 to donate £5, please help us to find a cure.

thebraintumourcharity.org

© The Brain Tumour Charity 2020. Registered Charity in England and Wales
(1150054) and Scotland (SC045081)



Registered with
**FUNDRAISING
REGULATOR**