

RASPBERRY Pi ZERO 2 W!

We test the all-new tiny Pi » Quad-core RP3A0 CPU

» Just £13.30 » Bluetooth 4.2 » 802.11n Wi-Fi » 40-pin GPIO

FREE DOWNLOAD
UBUNTU 21.10



LINUX FORMAT

PLUS!

- > FILESYSTEM ROUNDUP
- > MULTIPLE PI SCREENS
- > TERMINAL PODCASTING

The #1 open source mag

UBUNTU VS FEDORA

Discover who's the Linux grandmaster as the top distros try to checkmate the opposition!



67 pages of
tutorials
& features

Assemble car apps to drive AutoGrade Linux

Build your own Analytical Engine with open source

Interface key components with your Pi and Arduino

NOT SO OPEN
The trouble with modern licences

NETWORK SHOOTER
Adding multi-player features to your games



PAPER POWER

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

Discover the story of paper
www.lovepaper.org

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





LINUX FORMAT



» MEET THE TEAM

This issue we're comparing Fedora and Ubuntu, but what sort of distro do you favour? Something more LTS, something on a shorter update schedule or are you more a rolling-release type?



Jonni Bidwell

I use Arch, and also Arch32, Pop!_OS and Debian. And, albeit briefly, whatever's on our artisanally crafted DVD. LTS distros, through Snaps and Flatpaks, can now obtain the latest applications too, so adding hacky PPAs on the ilk of Mint and elementary OS is no longer de rigueur.



Les Pounder

My work machine runs Ubuntu 20.04 LTS. It gives me the stability to get my work done. For my other machines, my X220 is always distro hopping (currently Kubuntu 21.10) and my X390 is on Ubuntu 21.04. My Raspberry Pis are mainly using Raspberry Pi OS and Diet Pi.



David Rutland

I use Ubuntu 20.04 LTS which, among Linux distros, is the worn and comfy pair of solid old boots. Sure, I'll slip into Arch for an occasional dirty weekend, but it's like wearing high heels: super sexy so long as you don't slip and break your ankle (or distro). Comfort and reliability FTW!



Mayank Sharma

I prefer Manjaro's curated rolling release, which delivers updates a lot faster than the Ubuntu and the Fedoras, but isn't as maddening as a full-blown Arch. I'm not sure how Jonni manages it, but keeping pace with a full-blown rolling release is a bit too hectic for my taste.



Alexander Tolstoy

My main distro is the one with the green lizard, but I use both Fedora and Ubuntu from time to time. I tend to prefer Fedora because it successfully balances bleeding-edge software and stability. Moreover, most of the innovative desktop Linux additions that we use these days come from the Red Hat camp, which also backs Fedora.

Grandmaster clash



Ubuntu and Fedora are two of the big names in the Linux distribution world and their stewardships are set up quite differently. While they both have their funding foundations in large commercial segments of the Linux world, Ubuntu is specifically developed to be the commercial offering for all of Canonical's businesses: desktop (that happens to be used by home consumers); and its server, edge and cloud deployments.

Fedora, on the other hand, is the technological bleeding-edge development distro that's used to forge and test systems that will eventually find their way into Red Hat Enterprise Linux – the big serious distro that's created a multi-billion dollar business.

Let's not forget, of course, that Fedora is blue and Ubuntu is Orange! Luckily it's Jonni and Mayank rather than myself comparing the two projects to see what works well and what rubs them up the wrong way.

Approaches in how projects tackle installations should mirror users' needs. LTS builds offer stability, but at the same time provide software that's gradually aging (though Snaps and Flatpaks can work around this issue). In contrast, rolling release or more frequently updated distros mean constant updates, but greater security – at the cost of possible stability.

Rather than being critical of either direction, we need to bear in mind that it's all part of the varied open source ecosystem, which we're more than happy to embrace, because we can never envision one size fitting all. And mirroring that sentiment, this issue contains our usual wide range of features, reviews and tutorials. Enjoy!

Neil

Neil Mohr Editor
neil.mohr@futurenet.com



**Subscribe
& save!**

On digital and print –
see page 12

Contents

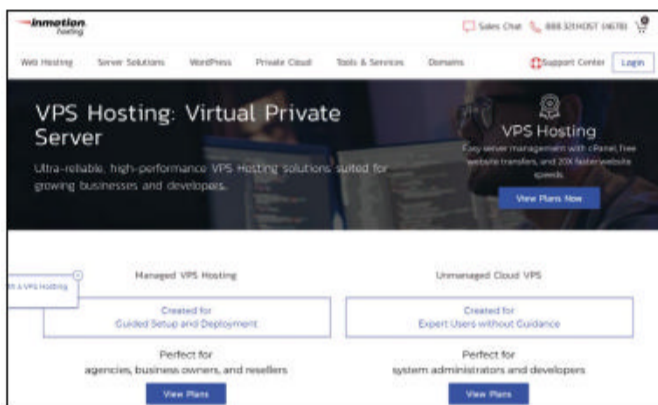
LAPTOP BAG OFFER!



SUBSCRIBE NOW!
Page 12

REVIEWS

InMotion Hosting 21
From dedicated servers to VPS you get feature-packed web hosting from a professional provider, says **Mike Williams**.



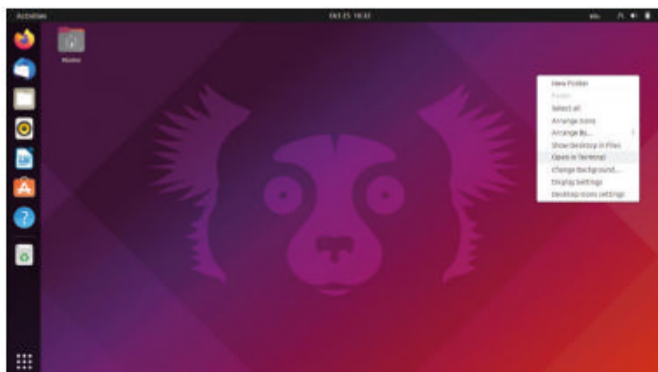
Atari Video Computer System 22
Monetising nostalgia should be an easy win for Atari so **Matt Safford** can't believe it's created another *E.T. the Extra-Terrestrial*.

Whonix 16 23
Mayank Sharma is a busy man, but when it comes to privacy and security, like Louis Armstrong, he has all the time in the world.

Devuan 4.0 24
The backlash against Systemd resulted in this sysvinit-friendly distro fork, now on to version 4 **Mayank Sharma** assesses its progress so far...

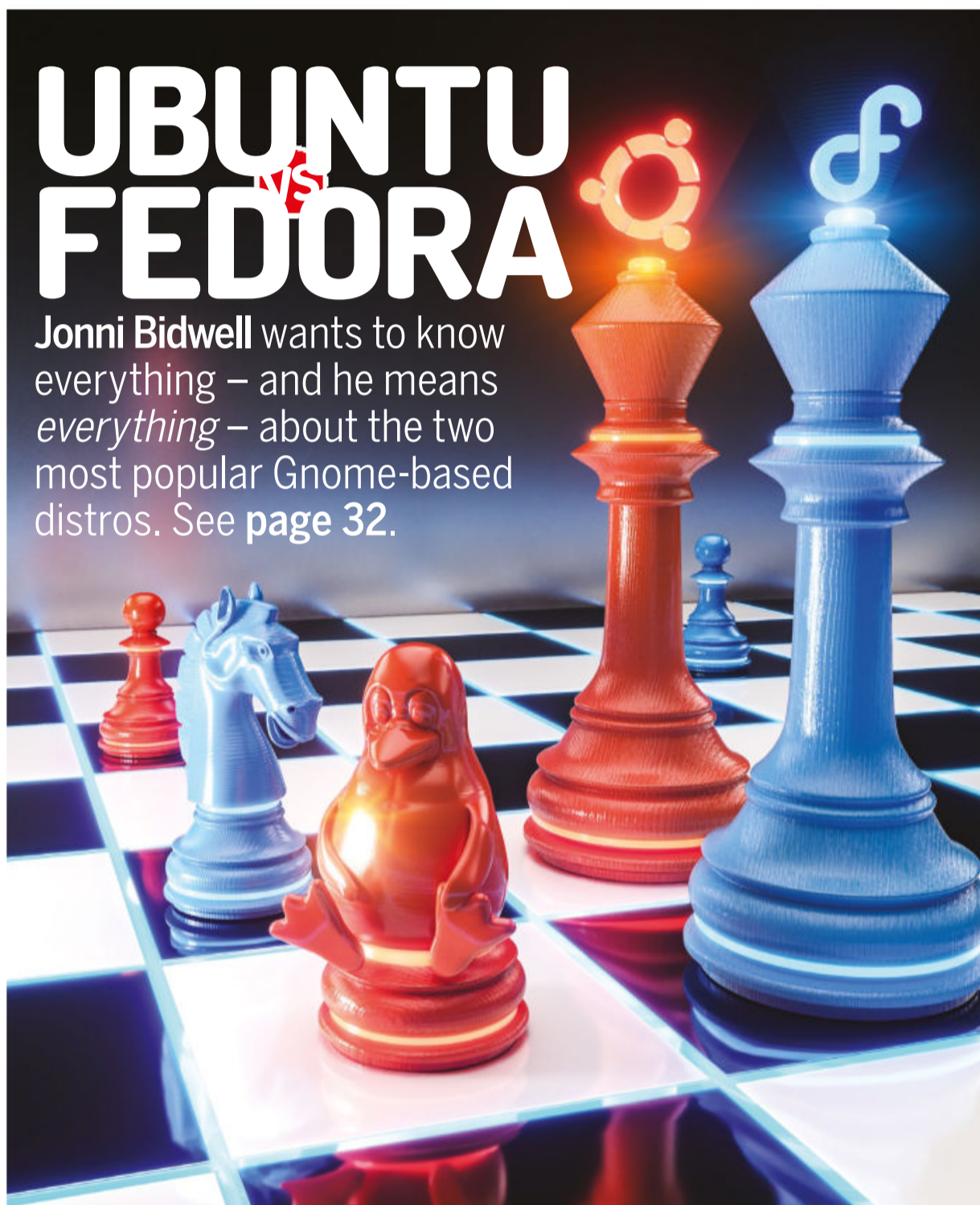
Zephix 3.0 25
A fan of portable, modular distros, **Mayank Sharma** runs across a relatively new one and wonders how it stacks up against his all-time favourite: Slax.

Ubuntu 21.10 vs Fedora 35 40
Mayank Sharma wonders if there's more that separates the two leading Gnome-based distros than their different packaging formats (*and logo colours!* – Ed).

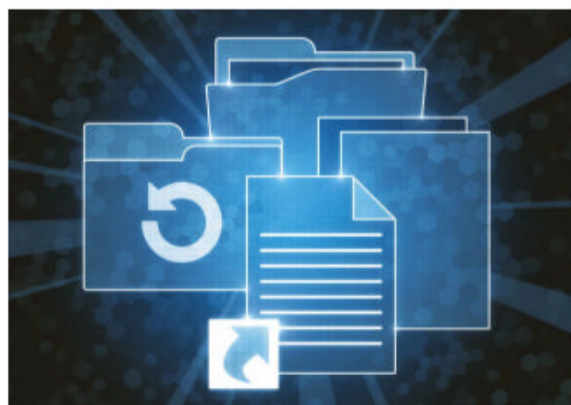


UBUNTU VS FEDORA

Jonni Bidwell wants to know everything – and he means *everything* – about the two most popular Gnome-based distros. See page 32.

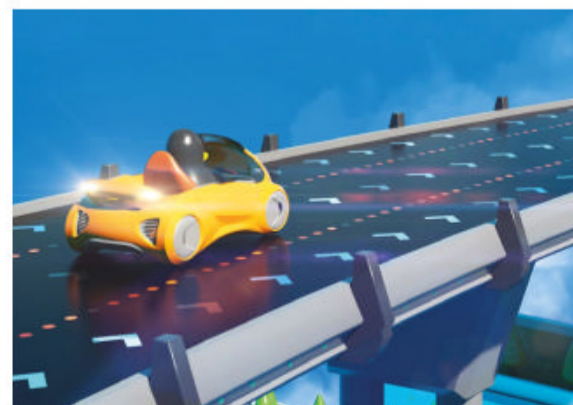


ROUNDUP



Filesystems 26
Alexander Tolstoy puts the major Linux (and some Windows) filesystems to the test and assesses which one is the best in terms of speed, reliability and extra features!

IN-DEPTH



Assemble AutoGrade Linux apps 42
Mats Tage Axelsson guides you through how open source frameworks and Linux can help you navigate the jungle of getting safely to your coding destination.

PI USER

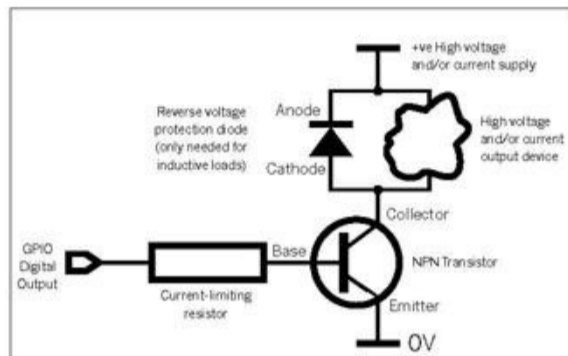
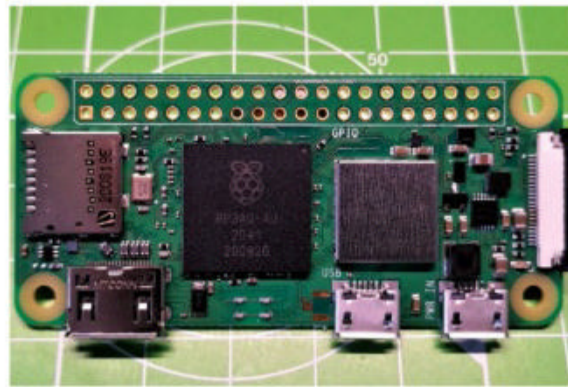
Raspberry Pi news 46
 Introduced by Michael Horn. The first price hikes for the Raspberry Pi.

Raspberry Pi Zero 2 W 47
 Les Pounder takes an early look at the latest Pi Zero model, with its updated CPU.

Advanced Scratch GPIO 48
 Les Pounder reveals how we can take Scratch further with a community-developed tool that builds on our Scratch knowledge.

GPIO electronic interfacing 50
 Mike Bedford shows you how to get more from your Raspberry Pi or other SBC by getting to grips with interfacing circuitry.

Multiple SPI screens 54
 Sean Conway explains how SPI works and uses it to set up a couple of displays.



CODING ACADEMY

Multiplayer Galaxians! 88
 Discover how to implement multi-player gaming in a *Galaxian*-style shooter with **Andrew Smith** and Python in the second part of coding *Star-Fighter*!

Develop a RESTful client in Go 92
 Develop a command-line client to access a RESTful server in Go with **Mihalis Tsoukalos** and the Cobra library.



REGULARS AT A GLANCE

News 6
 Google explains its kernel plans with Project Icebreaker, the Software Freedom Conservancy explains things to Vizio, the PinePhone Pro is explained and Fedora explains SilverBlue to us.

Kernel watch 10

Subscriptions 12

Answers 14
 How you compile projects from source again, how to get locked out from Mint 20, sorting grey scans out, sorting out UEFI entries and sorting out mail servers.

Mailserver 18
 Our dodgy torrents, more pinholes but in

space, why isn't Godot a Linux Foundation project and Linux for children.

Back issues 66
 Get hold of previous *Linux Format* editions.

Overseas subscriptions 67

HotPicks 81
Alexander Tolstoy hasn't got time to take a week off work and do nothing, he's too busy testing FOSS like: *Glyphtracer*, *OnlyOffice*, *Open Video Downloader*, *gThumb*, *JPEG XL*, *Annotator*, *Monitor*, *SpaceCadet*, *A/B Street*, *Metadata Cleaner* and *Fake Background*.

DVD pages 96

Next month 98

ON YOUR FREE DVD
 • Ubuntu 21.10
 • MidnightBSD
 See page 96



DVD pages 96
Jonni Bidwell guides you through getting to grips with Midnight BSD 2.1 and Devuan 4.

TUTORIALS

TERMINAL: Castero 60
Shashank Sharma has tried all the podcast clients and has finally found one of the best.

OKULAR: Editing PDF files 62
Nick Peers reveals all the open source tools you need to view, annotate, edit and create PDFs on your Linux-running PC. What else would it be running!?

EMULATION: Babbage Engines 68
Mike Bedford demonstrates how to program the mechanical computing devices that pre-dated modern electronic computers by over a century.

CHROME OS: Neverware install 72
 Enjoy a simple-looking operating system but with power, which can be added using a Linux container. **Matt Holder** is your guide to adding the Google OS to your system.

LXF SERVER: Run a Ghost blog 76
 Take control of your online content and install this highly popular lightweight and modular blogging system with a little help from the spooky **David Rutland**.

IN-DEPTH



Licence to Bill 56
Jonni Bidwell's studied licence minutiae and fears big business threatens the future of copyleft and software freedom.

Newsdesk

THIS ISSUE: Google's Project Icebreaker » Vizio sued over GPL wrongdoing » Linux on the PinePhone Pro » Fedora upgrade

KERNEL

Google to streamline its approach to the kernel

The tech giant announces Project Icebreaker, a new kernel project that could replace its aging Prodkernel.

At the recent Open Source Summit North America 2021 (<https://bit.ly/lxf283opensource summit>), Google engineers Andrew Delgadillo and Dylan Hatch announced Project Icebreaker, a new kernel project from Google that will be as close to the upstream kernel as possible.

According to Hatch, for every major upstream kernel release, Google will release a new Icebreaker kernel "on time, we want to stay caught up with upstream Linux." This will enable developers to introduce features that are close enough to the mainline that they can be included for the upstream.

This is one of the main problems with Google's current Prodkernel, which runs on data centres, because it's based on an older upstream kernel with a number of patches on top of it. The patches are rebased on to a newer kernel every two years, but during that time there are numerous changes to the mainline kernel, and this delay has meant that Google's developers can't easily participate with the upstream. Essentially, it's become its own beast separate from the mainline kernel, which causes problems for any developer who wants to propose a feature for the mainline when working with Prodkernel.

Delgadillo pointed out that this is an example of technical debt, and it's a problem that continues to grow worse, because every time Prokernel is rebased, the number of patches increases. In turn, that means the next time there's a rebase, it takes even longer – and this isn't sustainable.

Project Icebreaker should close the gap and enable upstream participation to happen much more easily, while also lessening the technical debt. As LWN.net reports (<https://bit.ly/lxf283icebreaker>), Hatch talked about how Google is aiming Project Icebreaker to be "a real production kernel," where arbitrary Google binaries could be run, and upstream changes could be validated ahead of a Prodkernel rebase. As soon as a new mainline kernel is released, testing can begin, rather than having to cram it in after two years before the next rebase occurs.

While Project Icebreaker sounds promising, there's plenty of risks involved, and it will require

GOOGLE HATCHES A PLAN

Project Icebreaker should close the gap and enable upstream participation to happen much more easily.

more feature branch testing. Dependencies will also need to be looked after carefully between features, which will result in more merges having to be performed. There's also the risk that if features in Icebreaker aren't included in the upstream, and more patches need to be added to Icebreaker, then it could end up going down the same path as Prodkernel and get further away from the mainline again. Google won't want to see history repeat itself when it comes to this.



Google's Prodkernel has fallen too far behind the mainline kernel, and it's causing problems.

LICENCES

Software Freedom Conservancy launches lawsuit against Vizio

TV manufacturer accused of GPL infringement.

The Software Freedom Conservancy (SFC) organisation (<https://sfconservancy.org>) is suing TV manufacturer Vizio over what it claims are “repeated failures to fulfil even the basic requirements of the General Public License (GPL).” Vizio, a popular TV brand in the US, uses a smart TV platform called *SmartCast*, which the SFC alleges “contain software that Vizio unfairly appropriated from a community of developers who intended consumers to have very specific rights to modify, improve, share, and reinstall modified versions of the software.”

While the SFC’s lawsuit isn’t looking for monetary damages, it “seeks access to the technical information that the copyleft licenses require Vizio to provide to all customers who purchase its TVs.” You can see the full lawsuit complaint at <https://bit.ly/lxf283sfccomplaint>.

According to the SFC, the non-profit organisation – which includes Debian Copyright Aggregation Project, Git, Wine and more – first contacted Vizio in August 2018, but by January 2020, Vizio stopped communicating with it. The SFC wants Vizio to share any source code it uses

for the *SmartCast* software “in a format that may be compiled without undue difficulty.” If this happened, the aim would be to help continue supporting Vizio TVs after the company no longer supports them with open source firmware. The open source community could also work on creating new firmware for these TVs, similar to how the OpenWrt project (<https://openwrt.org>) tackles embedded devices and routers.

According to the lawsuit, Smartcast uses code made by BusyBox, along with libraries such as ffmpeg and Systemd, which are all licenced under GPL. *SmartCast* is based on Linux, and that “multiple copies of the Linux kernel appear in the firmware.” It sounds like Vizio needs to ensure its software complies with GPL, and it’ll be interesting to see how it responds.



Vizio is accused of not complying with the GPL with its SmartCast software, used in smart TVs..

SMARTPHONES

PinePhone Pro announced

New details on the premium Linux-based smartphone.

PINE64 has revealed the PinePhone Pro, a high-end premium smartphone that runs Linux. According to the pre-order page (www.pine64.org/pinephonepro), the PinePhone Pro comes with a hexa-core Rockchip SoC running at 1.5GHz, along with 4GB dual-channel RAM, 128GB storage, a 13MP main camera and 5MP selfie camera.

Pine64 has worked closely with Rockchip when designing the PinePhone Pro to ensure that battery life is as long as possible, while also making sure the phone runs cool. For example, a bespoke suspend state has been developed that enables the PinePhone Pro to go into a low-powered mode to preserve battery, while still being able to receive calls and text messages. There’s also the same

privacy switches found in the original PinePhone, making it possible to easily turn off the webcam, microphone and more.

The design of the PinePhone Pro continues the premium feel. The IPS display is protected by Corning Gorilla Glass, the coating on the back is designed to minimise fingerprints, and the device retains the removable battery, which has also been upgraded to offer longer life between charges. It’s also compatible with existing PinePhone add-ons.

It’s shaping up to be an attractive phone, and developer pre-orders are live for \$399. You can find out more about this project at the PinePhone Pro launch blog at <https://bit.ly/lxf283pinephonepro>, which explores the hardware and has an interview with the team.

OPINION

TRAINED TO HIRE



Matt Yonkovit is Percona’s Head of Open Source Strategy and a member of SHA (Silly Hats Anonymous).

“ The Linux Foundation published its report on open source jobs last month. It revealed that 97 per cent of hiring managers said people with open source skills were a priority and 50 per cent said they would increase their open source hiring in the next year. Alongside Linux and cloud, demand for Kubernetes qualifications has skyrocketed by 455 per cent over the past two years. Many companies are encouraging their IT teams to use open source because it enables them to be flexible when dealing with problems, and on how they deploy and run their systems.

Clearly, the volume of roles requiring open source skills is growing. Two-thirds of developers say they need more training to carry out their roles. We don’t just need more training for people. We need better training and to increase the number of people with those skills. This puts the focus on creating, finding and keeping those people.

One reason for better training is to avoid teams having to “settle” for whoever they can get. Companies with under-skilled developers will find managing them time-consuming, and they may not be able to produce the right results at scale. Here, quantity is great, but quality is often better. ”

OPINION

MANAGED
CONTENT

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

“The CMS acronym, confusingly, can refer to either Content Management Systems, such as WordPress, or Configuration Management Systems such as Puppet, Ansible or Chef. For this issue’s column I’m using CMS to refer to the latter. There are a lot of them – indeed, Wikipedia lists 24 Open Source CMSs. If you have more than one Linux system then you should use a CMS, and they have a place even if you only have one Linux system.

There are many reasons to use a CMS, but here are just three to get you going.

First, it provides a change control mechanism and, when coupled with an SCM (Software Configuration Management), an audit log.

Second, it ensures consistency across your server farm, or subsets of it. Want emacs installed on all your servers, or how about modsecurity on all your Apache servers? A CMS makes that easy.

Finally, replicating or rebuilding a server is much easier if its configuration is stored in a CMS. Install Linux, install the CMS client, tell it to build a server of type X, and then consume coffee until the job’s done.

Once you manage your systems from a CMS, you’ll wonder how you ever managed without. ”

DISTROS

Is Fedora Silverblue the future?

Fedora Silverblue is envisioned as an image-based desktop OS alternative to standard Fedora.



Fedora Silverblue offers a novel way to install and set up an operating system.

Fedora’s Christian Schaller’s blogpost, titled *Fedora Workstation: Our Vision for Linux Desktop* (<https://bit.ly/lxf283fedoraworkstation>), discusses how Fedora Silverblue could be the future of Fedora. The Fedora Silverblue project (<https://silverblue.fedoraproject.org>) is a version of Fedora that comprises a single image based on OSTree and Flatpak, rather than hundreds of packages.

As Schaller explains, while many Linux distros are made up of multiple independently packaged libraries, “this setup is inherently fragile and requires a lot of quality engineering and testing to avoid problems, but even then sometimes things sometimes fail, especially in a fast-moving OS like Fedora.”

So, a distro like Fedora Silverblue would overcome many of these issues by being made available as a single image, and when it came to updating, the image would be replaced with a new one. This is achieved thanks to Flatpak, a universal app format that should make deploying applications in Fedora much easier, and OSTree, which Schaller describes as “a git for binaries, meaning it allows you a simple way to maintain and update your binary applications with minimally sized updates.”

While Schaller says, “We are not yet ready to declare [Silverblue] the official version of Fedora Workstation,” it’s clear the team is excited about what it’s achieved, and the possibilities it brings.

GAMING

Linux gets Anti-Cheat support

Epic is bringing its Easy Anti-Cheat software to Linux.

Epic Games has announced (<https://bit.ly/lxf283anticheat>) that it’s bringing its *Easy Anti-Cheat* software to Linux, and it will be free for all game developers to implement. This will help to reduce cheating in competitive online games. It should also enable a large number of games to be made available in Linux.

Valve’s *Proton* tool, which is based on *Wine* and enables Windows games to run in Linux, would often not work with some of the most popular games due to anti-cheat software not being available in Linux. However, as Epic explains, “Starting with the latest SDK release, developers can activate anti-cheat support for Linux via Wine or Proton with just a few clicks in the Epic Online Services Developer Portal.”



Epic Games move could bring a lot of new games to Linux.

WEB BROWSERS

Mozilla messes up new Firefox

Firefox 93’s launch blighted by an avoidable PR blooper.

Firefox 93 was recently launched by Mozilla, but even though it was positively received by critics, the launch was overshadowed by the discovery that there were now adverts included in the address bar, a drastic and invasive change that Mozilla had failed to properly disclose.

The feature, known as Firefox Suggest, started to appear for some users in *Firefox 92*, despite there being no official communication about the feature until a blog post on September 15 (<https://bit.ly/lxf283ffsuggest>), where it describes the feature as “a new discovery feature that is built directly into the browser.” Its inclusion in *Firefox 93* is hidden in the release notes, and it led to users complaining online.

The relative silence over the feature led to people worrying if data from their address bar was being shared, and while Mozilla clarified some aspects of the feature, for many people the damage had been done, and user confidence had been shaken.

Distro watch

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

MX LINUX 21

The latest edition of this Debian-based desktop distro has been launched. It comes in three versions with either Xfce, KDE Plasma or Fluxbox, and it's now built on Debian 11 Bullseye. Many applications have been updated: there's now installer partition options and a new UEFI live system boot menu has been included, which should make life easier for people who are using this as a live distro. Wi-Fi support has been improved, and Mesa Vulkan drivers are now installed by default. Find out more at <https://bit.ly/lxf283mxlinux>.



MX Linux is now based on Debian 11.

REDCORE LINUX 2102

Redcore Linux is a Gentoo-based distro, and the latest snapshot is available to download. Codenamed Polaris, this release brings new bug fixes, and while the release announcement (which you can read at <https://bit.ly/lxf283redcore>) admits that "you won't find many exciting new features inside [this release]", over 1,300 packages have been updated, and there's been a mass rebuild of the repository. Even if this isn't an exciting release, it's an essential one for Redcore users.



Redcore Linux 2102 brings important updates, but also showy new features.

UBUNTU STUDIO 21.10

Ubuntu Studio is a community version of Ubuntu that comes with the KDE Plasma desktop, and is aimed at creative professionals, especially in the audio and video production fields. The latest version, codenamed Impish Indri, is Ubuntu Studio's 30th release, and will be supported for nine months. The desktop has been updated to Plasma 5.22.5 with the Materia theme and Papyrus icons, and numerous applications have been updated. Check out the full release notes at: <https://bit.ly/lxf283ubuntustudio>.



Ubuntu Studio is aimed at music and video producers.

OPENBSD 7.0

OpenBSD is a UNIX operating system, and the latest version adds support for RISC64 architecture, as well as drivers for Apple M1 systems. This means that support for the likes of the new Mac mini and MacBook Pro 13-inch is improved, although the release announcement (<https://bit.ly/lxf283openbsd>) notes that it's "not ready for general use yet." Still, this is a step in the right direction, and hopefully solid Apple M1 support isn't too far off.



OpenBSD 7.0 brings some improvements to Apple M1 SoC support.

OPINION

PANFROST LEVELS UP



Alyssa Rosenzweig is a software engineer at Collabora, a global open source consultancy firm..

“ The open source Panfrost driver for Mali GPUs has now achieved official conformance on Mali-G52 for OpenGL ES 3.1. This is a big step forward for the open source driver, because it now certifies Panfrost for use in commercial products containing Mali G52 and paves the way for further conformance submissions on other Mali GPUs.

Conformance requires passing tens of thousands of OpenGL integration tests in a single run, and over the summer, we wrote hundreds of patches to fix failing tests. While no amount of testing can guarantee the absence of bugs, passing conformance gets us close. To ensure we remain conformant, we've upgraded our continuous integration infrastructure to run more tests before every merge. Ideally, we'd re-run the complete conformance suite for every commit, but that's not practical when a single run takes 11 hours.

Note the Mali-G31 and Mali-G72 use the same architecture as Mali-G52, so they should work well with Panfrost. Mali-T860 and older GPUs are not yet conformant, but still benefit from the driver quality improvements.

Most of the bug fixes have been backported to Mesa 21.2, so upgrade your Mesa and enjoy a more stable driver. ”

OPINION

HITTING
THE M1

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

“There was recently the first (but likely far from the last) Rust for Linux conference, known as “Kangrejos”. The purpose of the conference was to further work toward supporting Rust inside the kernel, specifically in driver code. Much interest in Rust exists within various communities because it simultaneously targets system programmers, but also aims to eliminate various memory safety issues inherent to C code. These issues arise from “UB” or Undefined Behaviour, and there’s a lot of it in the C programming language.

Rust aims to be both fast (and thus suitable for low-level programming) but also safe, since it provides careful tracking of the ownership of memory (pointer safety). For example allowing only one owner of an object, along with a mechanism (known as borrowing) for temporarily passing references between code functions. All of this is well and good, but I must confess that my first reaction to seeing talk of Rust inside the kernel was more along the lines of “here we go again”.

This is, however, what happens when you get set in your ways, and it can be a dangerous thing. I’ve been using Linux since the mid-90s, and as I enter my 40s soon, I seek to remind myself that I don’t want to be “that guy” in the rocking chair yelling at the kids to get off my lawn. It’s time to play with Rust and see what the fuss is about. I’ll write up my adventures with Rust in a future issue. In the interim, readers may enjoy Paul McKenney’s new blog series on the topic (paulmck.livejournal.com), beginning with adventures in the memory model.

KERNEL WATCH

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

Linus Torvalds has released what will likely be the final RC (Release Candidate) kernel prior to 5.15 final.

The new kernel will contain a number of features that we’ve covered previously, including a new NTFS filesystem driver, and a new in-kernel “KSMBD” Windows SMB3 network filesystem server intended to compliment the well-known Samba userspace based server. However, there are those who question why the kernel needs a new in-kernel driver vs using Samba instead. More importantly, a number of security issues have already been found in the KSMBD driver, leading to questions of quality.

With the imminent release of 5.15 will soon come the merge window for what will become 5.16 in the early new year. We’ll have a complete summary of the features that are merged in the next issue. Meanwhile, it’s worth giving a quick acknowledgement to the members of the Linux Foundation Technical Advisory Board (TAB), who were all re-elected to serve another term. The five members provide guidance to the Linux Foundation on sensitive matters, and are also accountable for handling violations of the code of conduct.

Page folios

One feature that will probably not be merged into Linux 5.16 is the folios work by Matthew (Willy) Wilcox. It’s hit a few roadblocks during the review process that led Linus to saying, “I’m not merging something core like this while it’s under active discussion.”

Folios attempt to clean up how the kernel manages compound pages. In a typical PC, a CPU’s memory management unit (MMU)

handles memory in units of 4K (or sometimes 16K or 64K). Software constructs “page tables” that are “walked” by hardware in translating a virtual address to underlying physical hardware addresses. These page tables can group pages into larger units, known as huge pages, that are multiples of 4K. There are other occasions where the kernel wants to deal in larger chunks, such as when allocating contiguous buffers for DMA (direct memory access).

The kernel traditionally represents such larger “compound” pages as a “head” page followed by a series of “tail” pages. You can perform various operations on a tail page, and in many cases the kernel will redirect those to the associated head page. But not every case. As Willy put it, “A function which has a strict page argument might be expecting a head or base page and will BUG if given a tail page”. His idea is to create a “page which is guaranteed not to be a tail page”. Beyond robustness, there are ways in which folios may improve performance, and even facilitate memory compaction later.

It sounds straightforward enough, but the implementation of folios has involved digging into the interface between filesystems and the “page cache”, a region of memory used to cache data coming in from disk. A variety of different APIs exist, but not all of them are friendly to conversion. The topic has festered on the kernel mailing list for months in the form of rounds of RFC (request for comment) patches, but did finally come to a head with a discussion at the 2021 Linux Plumbers Conference and subsequent emails. Still, it’s not clear where things are going to end up. Linus didn’t hate the idea, yet others have strongly negative opinions. We shall see what happens. **LXF**

» ONGOING DEVELOPMENT

Various updated patches for Apple Silicon “M1” Mac support have been posted, including version 4 of a series adding PCIe, and assorted other IP blocks on the M1 SoC. One for the PASemi i2c driver targets a fairly mundane piece of logic, but it comes with a summary of the history of the hardware, and how it arrived in the M1 via the acquisition of P.A. Semi that Apple made back in 2008. As the code notes, the hardware “hasn’t

changed much since then”. A lot can actually end up staying the same between chips.

The much-delayed support in mainline Linux for virtualisation on RISC-V systems may finally have a path forward, as the “H” ISA extension providing the machine instructions has been ratified by RISC-V International. Anup Patel posted updated patches for consideration, which were then ack’ed by RISC-V kernel maintainer Palmer.



SPORT



TECH



The Perfect Christmas Gift

from just £6



MUSIC



HOBBIES

SUBSCRIBE NOW



- Give the gift that keeps on delivering
- Over 80 magazines to choose from
- Free delivery straight to your door

www.magazinesdirect.com/xm21

Terms and conditions: Offer closes 31/12/2021. Savings are based on the newsstand cover price. Please allow up to six weeks for delivery of your first subscription issue in the UK, and up to 8 weeks for overseas. All gift subscriptions will start post-Christmas with the next available issue. Payment is non-refundable after the 14 day cancellation period unless exceptional circumstances apply. For full terms and conditions, visit www.magazinesdirect.com/terms. For enquiries please email help@magazinesdirect.com.

The Perfect Christmas Gift

for just **£55.99**

plus

**FREE! STM MYTH
LAPTOP BAG**

**WORTH
£90!**



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

The remarkable CableReady system is an incredible tech innovation that sets STM apart from other bags. It enables you to charge on-the-go with lengths of cables out of sight.

The air channel back-panel-and-yoke system is cleverly designed to wrap around you and provide improved weight distribution. In other words: maximum comfort.

Your special Christmas offer

- Free STM Myth Laptop Bag worth £90
- Enjoy 12 months of Linux Format
13 magazines for just £55.99 – that's just £4.30 an issue!
- Receive every issue delivered direct to your door
- Get full LXF Archive access with PDFs back to issue 66 (2005)
- A thoughtful gift that keeps delivering throughout the year



Upgrade to include digital for only £10

- Instant access any time, anywhere
- Never miss an issue of Linux Format magazine
- Available on iOS or Android

Order the gift that keeps on delivering

WWW.MAGAZINESDIRECT.COM/LIN/XM21

or call **0330 333 1113** and quote **XM21**

Answers



Neil Bothwick

takes two Tuxes into the bathroom, as they're penguins.

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

Where is the program?

I read about *PDF Mix Tool* in LXF281. This is something that I have always wanted to have, so I downloaded the file `pdfmixtool-master.tar.gz` and then extracted it as `pdfmixtool-master`. This is a directory containing some files. My problem now is, how do I start the program? I am baffled.

Bryan Mitchell

The file you downloaded is the source code from the project's GitLab repository. All open source software is distributed in this way, although much of

it is also built into packages by the distro maintainers. The steps for installing from source can vary, although there are standard ways of doing it. There is normally a README or INSTALL file within the files you unpacked that contains instructions for compiling and installing the software. In the case of *PDF Mix Tool*, it uses the CMake build system. The first step is to make sure you have CMake installed – check in your package manager. While you are there, also see that you have Qt, *Qpdf* and *Imagemagick* installed, these are needed when building and running *PDF Mix Tool*. Now open a terminal and change to

the directory containing the files you unpacked and run these commands to build the software.

```
$ mkdir build
$ cd build
$ cmake .. -DCMAKE_INSTALL_PREFIX=/usr/local -DCMAKE_BUILD_TYPE=Release
$ make
```

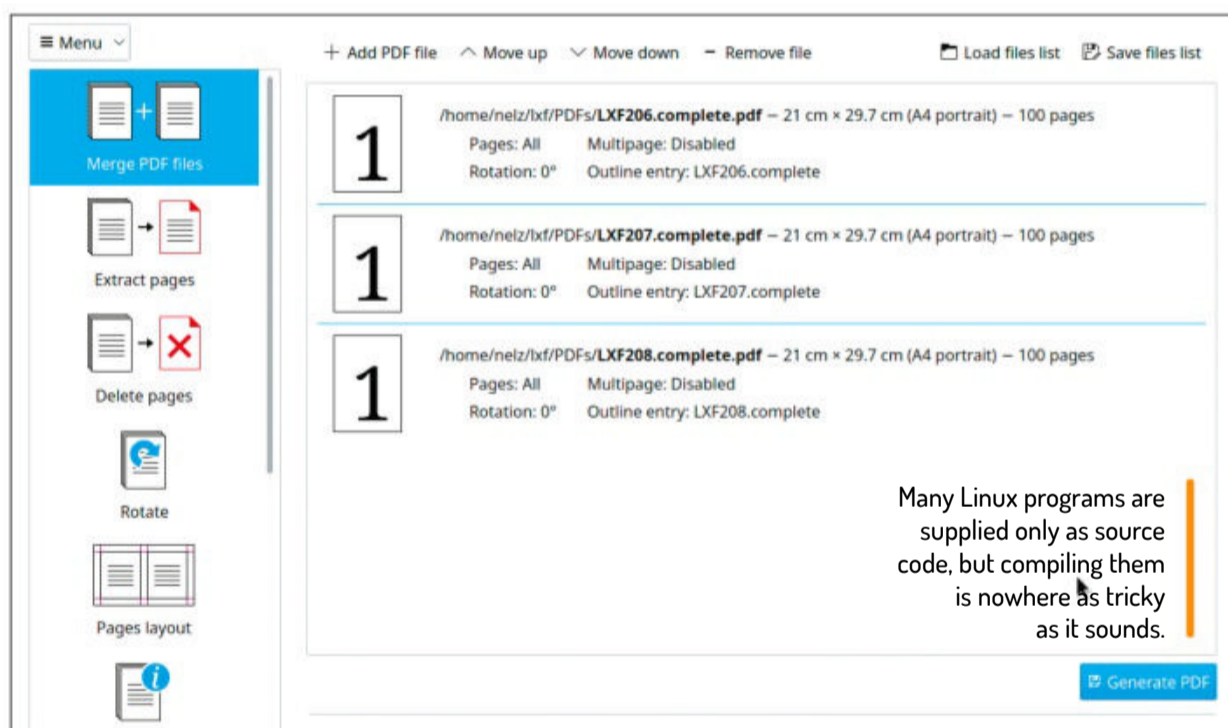
Once this has run, you will find the executable file in the `build/bin` directory. This is good for temporary testing, but if you want to make it permanently available, you can install it with

```
$ sudo make install
```

When the compilation is carried out as your normal user, all files are written to the `build` directory so no special permissions are needed. But installing to the system directories requires root permissions, hence the use of *sudo*.

This approach, or something similar, works with the vast majority of open source software out there and it becomes quite straightforward once you have done it a couple of times. In the case of *PDF Mix Tool*, there are other options as it has been packaged in Flatpak and Snap formats. These are distro-independent packaging systems that allow you to install the software with one of:

```
$ flatpak install flathub eu.scarpetta.PDFMixTool
$ sudo snap install pdfmixtool
```



» A QUICK REFERENCE TO... UEFI

UEFI (Unified Extensible Firmware Interface) is how x86 computers are booted nowadays. Instead of having to cram all the initial boot code into around 400 bytes at the start of the disk, UEFI uses a FAT-formatted partition, usually the first partition on the first disk. Known as the ESP (EFI System Partition), this contains one or more bootloader files. These files can load a full bootloader like *GRUB*, a compact boot manager like *rEFInd* (www.rodsbooks.com/refind) or load your Linux kernel directly.

Left to its own devices, the firmware will load the first (or only) boot option. Or you can hold down a key to see all options, including entering firmware setup (what we used to call the BIOS menu) where you can edit the various boot options.

Most UEFI firmwares include a CSM (Compatibility Support Module) mode that can boot legacy BIOS bootloaders – normally only needed if you are booting something like a live distro that is not UEFI-enabled. Only use this when you

have to as booting an install disk with CSM means that the installed system will not be UEFI-enabled.

In Linux, you can also add, remove and reorder boot entries using the *efibootmgr* command line tool. When using *efibootmgr*, do bear in mind that the EFI spec requires DOS-style paths, with backslashes instead of slashes. As the backslash is a special character in Linux shells, not only do you have to convert paths to DOS notation, you also have to quote them.

You will need the Flatpak or Snap software installed on your computer, and you can do this in your package manager.

Q Locked out

I recently installed Linux Mint 20.2 on my system. All was going well and I set up a Samba share and added my account into a group that I use to access the Samba share. Later on I restarted my Linux Mint system and now I seem to be removed from the root/sudoers group. I don't have any other accounts with root access on the system and when I try and run `sudo usermod -aG sudo <my username>` I get the following error: Sorry, user <USERNAME> is not allowed to execute '/usr/sbin/usermod -aG sudo <USERNAME>' as root on Linux Mint.

Didier Rouleau

A As you seem to have lost the right to use `sudo`, there are two possible causes. The first is that the `/etc/sudoers` file has been altered or corrupted. You cannot read this file without root access so this is a little difficult to check, but you can list it to see when it was last modified:

```
$ ls -l /etc/sudoers
```

The other possibility is that you are no longer a member of the group required to use `sudo`. On some distros this is the `admin` group, on some it is the `wheel` group (another hilarious joke going back to the ye olde days of Unix) and on some distros, including Mint, it is the `sudo` group. You can see which groups you belong to by running `groups` in a terminal. This is the more likely cause given that the problem appeared after you manipulated your user's groups. You probably replaced the list of groups with the one needed by the share rather than adding that group to the list.

This is easy to fix by editing `/etc/groups` if you have root access, which you can get in one of two ways. The first is to boot into rescue mode, and at the `GRUB` menu (press Space when booting if this is not shown by default), press E to edit the boot option and add `systemd.unit=rescue.target` to the end of the options. This should boot you into single user mode, from which you can edit `/etc/group`. There should be a line like

```
sudo:x:27:
```

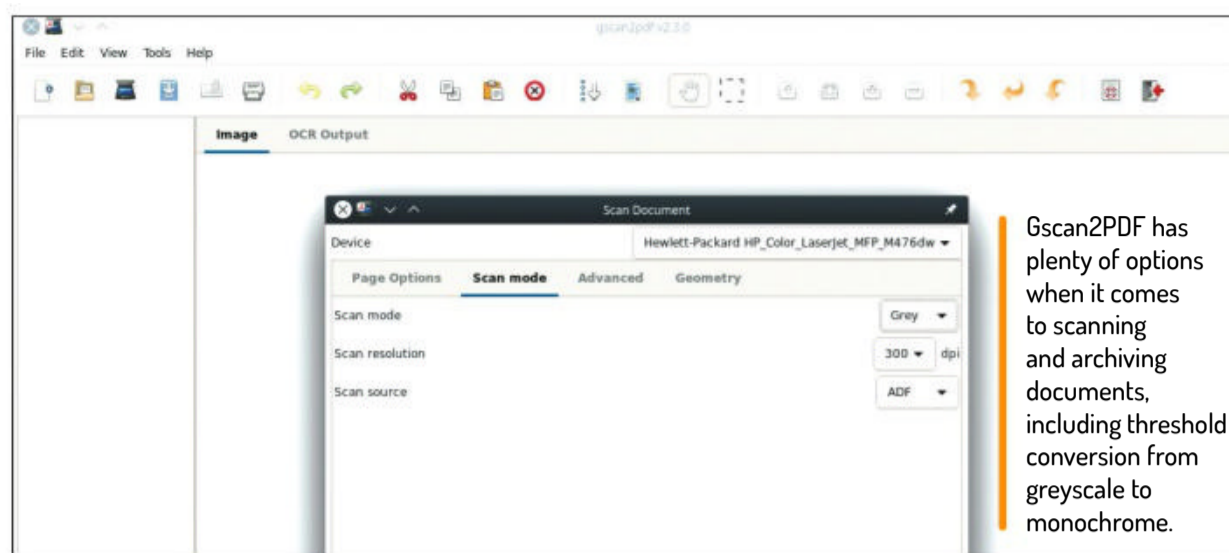
The number may be different. Add your username to the end of this line:

```
sudo:x:27:myuser
```

and save. If you get a message about the file being read-only, your root filesystem has probably been mounted `ro` when booting, so run:

```
$ mount / -o remount,rw
```

and try again. If this still fails, boot from a live distro, such as the Mint installer, and



mount your root filesystem. You can see which is the root partition by running:

```
$ findmnt /
```

while running the distro normally. Then boot the live distro, open a terminal and:

```
$ sudo mkdir -p /mnt/root
```

```
$ sudo mount /dev/sda1 /mnt/root
```

```
$ sudo nano /mnt/root/etc/group
```

Replace `sda1` with your root partition. Make the group change as above and press Ctrl+X to save and exit. Reboot and you should have `sudo` rights back. Now you just have to find out which other groups you should have been a member of and are no longer.

If it is a corrupted `sudoers` file, the simplest repair is to boot the Mint live distro and mount your root filesystem as above, then copy `/etc/sudoers` from the live system to your root filesystem.

Q Fade to grey

My Canon Canoscan LIDE scanner outputs PDFs of scans of pages of text as black lettering on a dark grey background. Is there any GUI software available that can quickly and easily threshold the black to black and the grey to white? If this cannot be done with PDFs, I could convert the PDFs to JPEGs. The last time I did this it was a multi-stage process that took all afternoon.

Tom North

A You don't specify which software you are using; they generally have an option to scan in colour, greyscale or mono. Have you tried scanning in mono? However, if you are getting a white background appearing as dark grey, it is possible that a mono scan may just give an all-black page.

We like `gscan2PDF` as a scanner GUI as it provides a great deal of control over the scanning process. It also has a handy threshold function for converting greyscale scans to monochrome. This function allows you to set the threshold level, so it should be easy enough to determine the correct value to render your dark grey

backgrounds as white with a couple of trial runs. Because it works on the data in memory after the scanner has done its job, it is a simple matter to try a value then hit Undo and try another until you get the result you want. You can also load existing PDFs to process them in the same way.

`Gscan2PDF` has a number of other useful features, and can save in a lot more formats than PDF, despite the name; we find DJVU excellent for reducing file sizes without affecting image quality. It also has the option to OCR scans of text and store the text as a separate layer, creating PDFs that can be searched.

Q Double Debian

I have two Debians marked in the BIOS, one probably from some previous installation or maybe Ubuntu, and the other is the one currently installed.

Foreseeing that this may bring me some confusion in the future I would like to remove one from the BIOS, the one that is not installed on the system but marked as existing. How can I distinguish which is the authentic one from the fake one to remove the latter?

Peter Shaw

A Are you using UEFI? It sounds like it, so strictly speaking you have no BIOS, just UEFI firmware. This is a fairly common problem, caused by distro installers hard-coding the label. Anyway, you can list the boot entries from the firmware with:

```
$ sudo efibootmgr
```

This will list the entries, but not give much detail. Add `-v` to the command and it will show you much more about each entry, hopefully enough to determine the difference between them. Once you know which one you want to get rid of, you can do it with:

```
$ sudo efibootmgr --bootnum NNNN --delete-bootnum
```

where `NNNN` is the number of the boot item you want to delete. Alternatively, you could rename one of the Debian entries,

making it easy to distinguish between them and try each one to see which is the one you want. Unfortunately, *efibootmgr* can only set the label on new entries it creates, it cannot change existing entries. The only way to rename is to delete and recreate an entry with a new label. This scary-sounding process has been made easier with a script available from <https://github.com/s-n-ushakov/rename-efi-entry> – download this and set the executable bit. Then you can rename an entry with:

```
$ sudo ./rename-efi-entry.bash debian
Debian2 NNNN
```

Note that, unlike *efibootmgr*, you must give the full four digits of the boot number. You give the old label, the new label and the boot number in that order. If either of the labels contain spaces you should enclose them in quotes. We have tested this script and it did just what it was supposed to do. Because of the way this script works, deleting and creating a new entry, the boot number will change, and possibly also the boot order. Once you are happy with your boot entries, you can use *efibootmgr* to set whichever order you like:

```
$ sudo efibootmgr -o NNN1,NNN2,NNN3...
```

The entries should be separated by commas with no intervening spaces. You do not have to list all the entries, only those that you want to order.

No print Mint

Linux Mint 20.2 is

generously reviewed in LXF281 with a recommendation that all users should upgrade. I was running 19.3 on both my desktop PC and a ThinkPad notebook. When version 20 appeared I decided to try it on the notebook for a start and managed to upgrade to 20.2 in January.

The changeover went well and all seemed to be normal, except that I was now no longer able to print to a Canon Pixma inkjet, a HP LaserJet and Boomaga despite loading drivers. There had been no problem whatsoever with 19.3 which, for obvious reasons, I am still using on my desktop. If there is some mysterious workaround required for 20.2 that works, I would be delighted to hear of it so that I can also upgrade the desktop.

Ken Arntsen

A When you say you cannot print, what happens? Is the printer not available? Does it show up in the list of printers but anything you send to it just disappears, or do you get some sort of error message? The first thing to check is whether the printer is seen, by running:

```
$ sudo lpstat -s
```

in a terminal, or by looking at the printer list in the CUPS web interface at <http://localhost:631>. If the printer is present, you can see what happens when you try to print by examining the logs at `/var/log/cups/error.log`. You can watch this log while trying to print by running:

```
$ tail -f /var/log/cups/error_log
```

This is not a generic problem with Linux Mint 20.2; if it were all hell would have broken loose long ago and the problem fixed. You just need to do a little detective work to gain an idea of what is happening.

Mail without an MTA?

I use a cheap Debian computer as a NAS to back up important files daily over the LAN. I'd like to add a *cron* job so that the NAS checks disk usage, and sends an email the day the disk is 80% full. Is there a way to send an email

without setting up a whole MTA server like sendmail, Postfix, etc?

Robert

A There are a number of ways to do this. You could have your checking script talk directly to an SMTP server at your ISP. How you do this depends on the language you choose for your script.

Alternatively, you can install a minimal mail server. Unlike *Postfix* et al, these simply forward emails to a server elsewhere. Examples of these include *ssmtp* and *nullmailer*. Going with *nullmailer*, you need to configure the various files in `/etc/nullmailer`, the most important of which is called **remotes**. This file contains a number of commented examples, but for a straightforward SMTP setup you only need one line:

```
mail.myisp.com smtp
```

The **remotes** file must not be world-readable. *Nullmailer* runs as a daemon, so you need to make sure it is always running. If your distro uses Systemd, these two commands will start the daemon and make sure it is fired up every time you boot:

```
$ sudo systemctl start nullmailer.service
$ sudo systemctl enable nullmailer.service
```

and `$ sudo systemctl status nullmailer.service` will show you if it is working and any error messages or warnings. You may also need to issue a restart command after editing the configuration files. Once *nullmailer* is running, you can use the standard mail command to send mail:

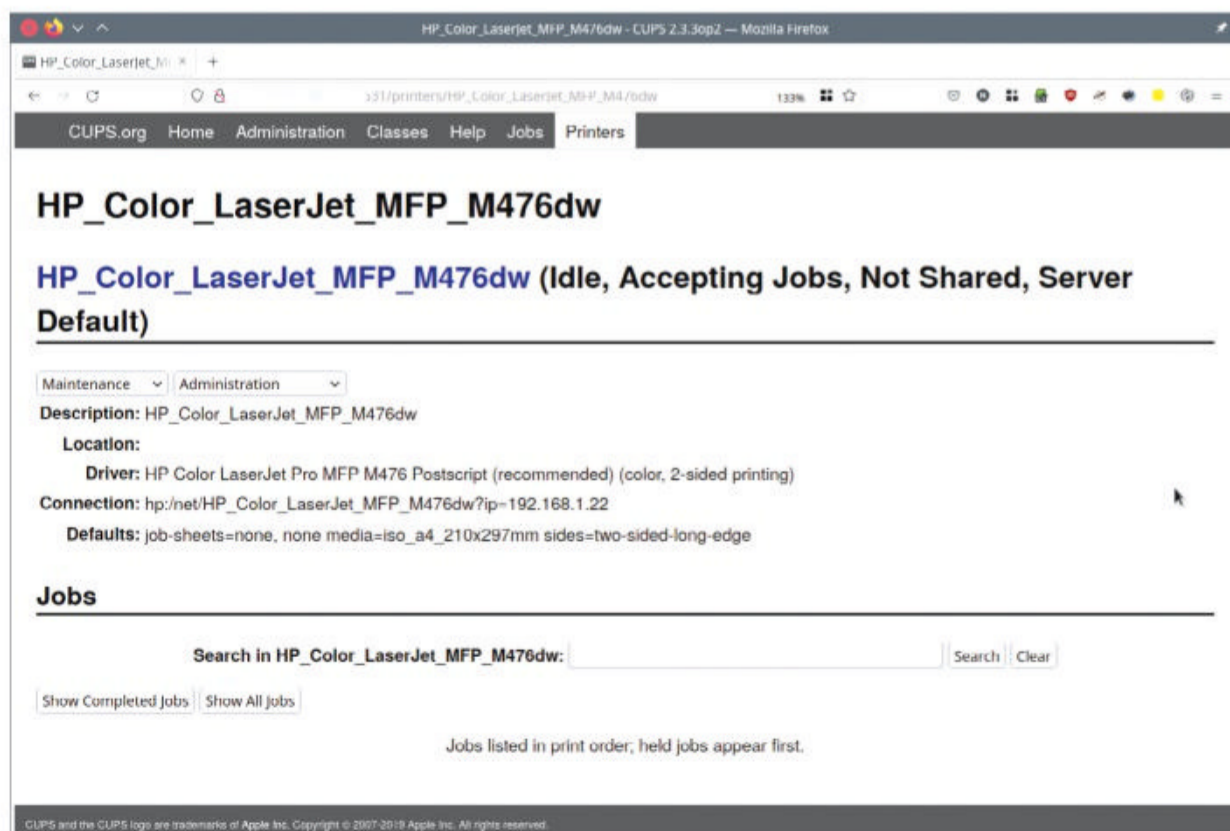
```
$ echo "mail body" | mail -s "Message subject" you@email.com LXF
```

GET HELP NOW!

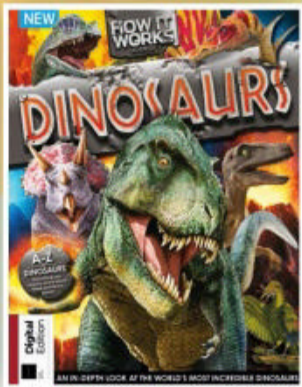
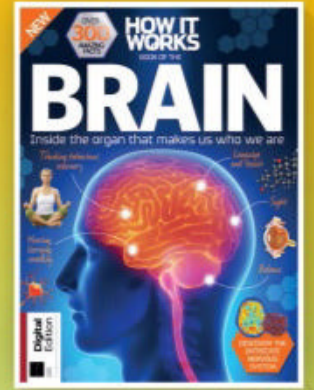
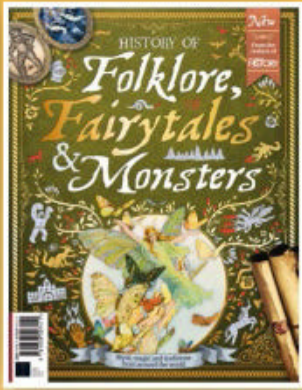
We'd love to try and answer any questions you send to lxf.answers@futurenet.com, no matter what the level. We've all been stuck before, so don't be shy. However, we're only human (although many suspect Jonni is a cyborg), so it's important that you include as much information as you can. If something 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
```

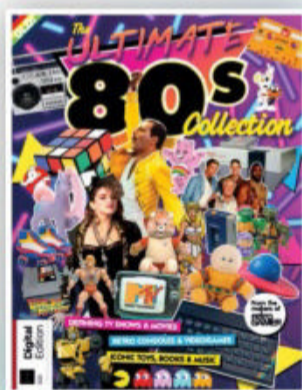


The CUPS web interface is easy to use, but sometimes you need to get down and dirty with the error log.



DISCOVER OUR GREAT BOOKAZINES

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



Follow us on Instagram  @futurebookazines

www.magazinesdirect.com

Magazines, back issues & bookazines.



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.

Pinholes in space

Thanks for the nice article on pinhole cameras in LXF281. I am an astronomer and, believe it or not

I used a pinhole camera when first checking out my specialist CCD camera before attaching it to the telescope. The images from the pinhole prove that the camera and its software are working correctly.

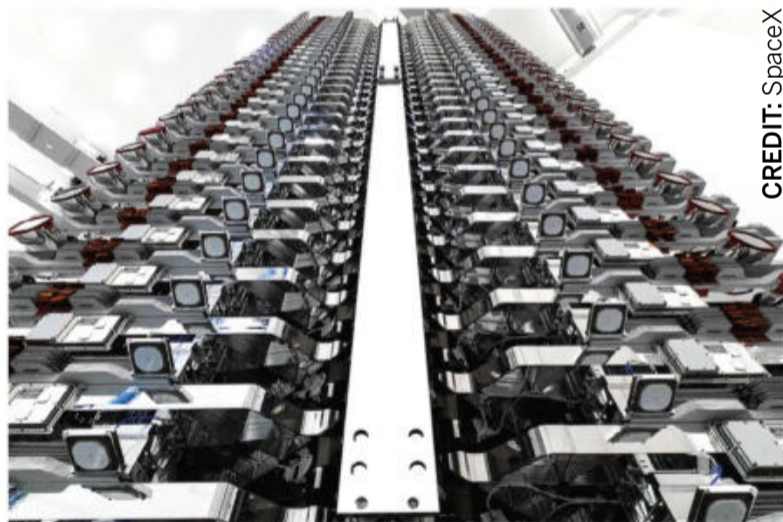
Astrophotographers, especially those who wish to make precise measurements of their images, routinely have to deal with dust specks, vignetting, detector non-linearities, and so on. They do this through a process called flat-fielding. First, one takes an image of something which ought to be featureless, such as the twilight sky or a lightbox, or even an out-of-focus monitor showing a white screen. That image, the flat field, will not be blank but will show the aforementioned blemishes.

Thereafter, any subsequent ('science' or 'light' in the astro jargon) image is divided by the flat-field image to remove the blemishes and the result used for subsequent processing. Almost any image processing software, such as the *GIMP* or *ImageMagick*, will let you do that with ease. Image restoration, including sharpening out-of-focus images and removing motion blur, is another fascinating topic but too complex to get into here. A future article perhaps?

Paul Leyland

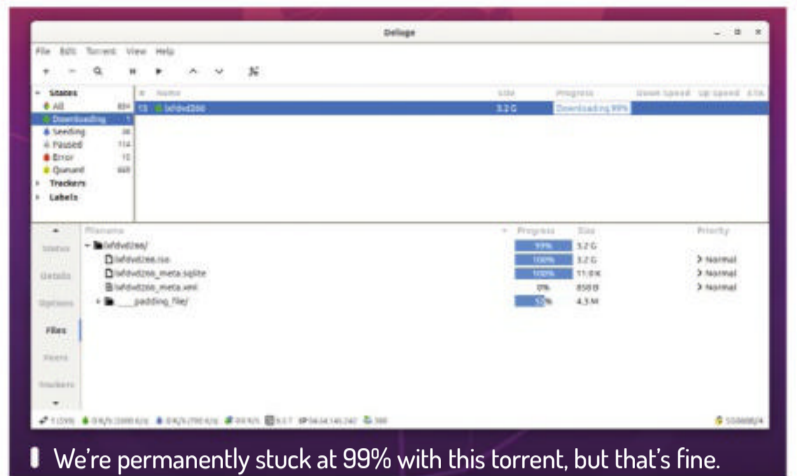
Neil says...

That's certainly an eye opener. I'd love to do more astronomy tutorials... perhaps I should start hanging around our space.com colleague's desks, talking loudly about Linux. I'm sure that's not going to annoy anyone!



CREDIT: SpaceX

SpaceX's first 60 Linux-running Starlink satellites stacked flat atop each other for launch in May 2019 by a Linux-running rocket.



ISO trouble

Is there a problem with the LXF281 ISO? Tried to download it a few times over the last couple weeks with *Free Download Manager* but it sticks at 99% (2.14GB)/2.15GB. In case it was a problem with *FDM* I downloaded qtorrent this morning and the issue is the same.

Nick Brooks

Neil says...

This appears to be an odd 'bug' where padding files that are added alongside the ISO file don't complete. So if you check the torrent status the main ISO will be downloaded 100 percent, but the problem is with the padding files. It's annoying but doesn't stop you getting the full ISO itself.

Godot out

In the LXF280 Newsdesk you discussed the Open 3D Foundation and its first project, the Open 3D Engine which is based on the Amazon Lumberyard engine. Where does Godot fall into this, and why is it not a flagship for O3DF – or at the least the basis for the O3DE?

Chris Williams

Neil says...

That's a good question and you'd have to ask the Linux Foundation for the absolute reasoning here, but it seems this was more Amazon asking the Linux Foundation to help set up the Open 3D Foundation, rather than the Linux Foundation shopping around for likely candidates. You can imagine it helps establish Amazon's Lumberyard as a more attractive proposition for developers, while Amazon will bring a level of commercial gravitas with its additional cloud services.

Helpdex



CREDIT: Godot



We're pretty happy Godot is doing its own thing.

As much as we love Godot it's certainly more of an indie development that might not offer the range of tools Lumberyard can, especially as it's being aimed at the film industry too.

Child Linux

I'm struggling to set up Linux for my bright grandson who's into chess. His parents don't want to encourage him to play mindless games and have open access to the internet, so what's the best/easiest way to do it?

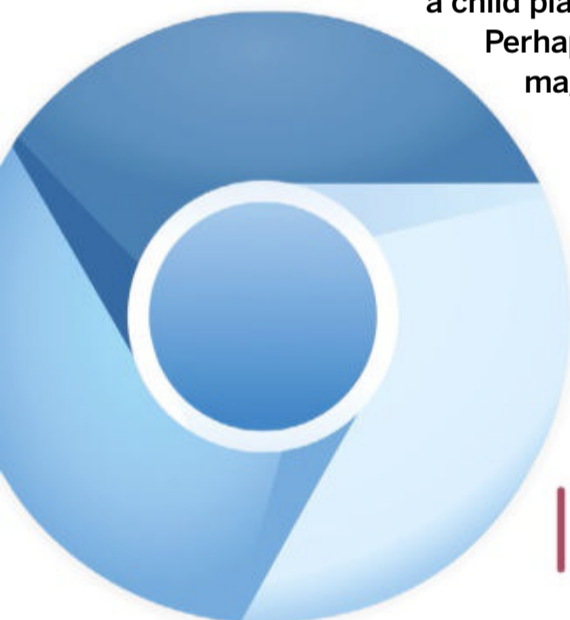
Firstly, I can either give only his parents root access on a distro, which has pros and cons as his parents won't want to have to keep going back and forth to do updates. Or I can give him a live distro with persistence on a USB stick, which might work – but which distro? Most of the educational Linux distros seem to be no longer supported. He doesn't need anything too complicated or liable to self-destruction even if he is pretty bright, but he's not a child playing with toys either.

Perhaps a project for a magazine to cover?

I think a Linux computer is undoubtedly a superb educational tool – much better than a PlayStation, Xbox or Nintendo!

Philip Cooper

Many schools are using Chrome OS for helping educate children, so it's not an awful choice.



» LETTER OF THE MONTH

We did that!

Microsoft is again trying to control the direction of travel by imposing hardware compliance, ie. TPM 2.0, for the roll-out of its Windows 11 OS that started in October.

The issue that needs to be explored is whether this will impact Linux, and if so how and when? Information on the purpose of the add-on chip and its effect on boot-up and running performance needs to be explored, as well as whether such 'enhancements' add value to overall system security. Given that Windows 10 is supposedly Linux-compliant with its WSL subsystem, the impact on the Linux community could be important.

John Ireland

Neil says...

Hopefully then last issue will have cleared up a lot of these questions – go, us! I'd say it feels like the Secure Boot issue all over again, but perhaps even less of a headache. As Jonni explained, the TPM itself isn't a problem; it's whether hardware manufacturers enable the correct settings to disable the parts required to enable other operating systems to boot. This was the issue with Secure Boot and will be the issue with the TPM.

Linux users are going to have to be savvy enough to check that new PC systems provide alternative OS support before buying, though the wider industry issue is enabling computer equipment to be used for longer, rather than being locked to potentially unsupported old releases of Windows 11 as Microsoft drops support for them.

Perhaps 'Linux loves Windows' is going a bit far... 'tolerates'?



Neil says...

This is a really good question and I think it is something we need to cover down the line. I was pondering the same issues a while ago and ended up installing a build of Chromium OS on a system, so that my daughter could access her school's Google Docs without being able to willy-nilly install anything else – but it's certainly not what I'd call an ideal solution. **LXF**



shane_collinge@yahoo.com

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!



ON SALE
NOW



Ordering is easy. Go online at:

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

Or get it from selected supermarkets & newsagents

InMotion Hosting

Feature-packed web hosting from a professional provider, says **Mike Williams**.

IN BRIEF

We like the features, the support, the 90-day money-back guarantee, but the best part of InMotion Hosting is that it gives the solid impression of a reliable, professional, honest web-hosting company that isn't trying to rip you off.

InMotion Hosting is a popular US-based web-hosting provider with a plan for just about everyone: shared hosting, a WordPress and BoldGrid-based graphical Website Creator, managed WordPress, managed and cloud VPS, reseller plans, managed and bare metal dedicated servers, high-end enterprise products... the list goes on.

The low-end shared hosting plans have some significant limits, particularly with disk space, email accounts and databases. For example, the Lite plan offers unlimited bandwidth, a free domain, free SSL and cPanel site management. But you're restricted to 10GB storage, two databases and a single email account with only a 100MB inbox, all for \$2.49 a month over three years, \$7.49 on renewal.

The higher Launch, Power and Pro plans are mostly about lifting the limits (databases, websites supported, subdomains), improving performance, adding additional e-commerce support and (for the top-of-the-range Pro plan) a 99.9 per cent SLA. Designed for small businesses, the 50 website, 100GB storage, 50 email accounts Power plan costs from \$4.99 a month on the initial term (\$13.99 on renewal). If you're looking purely at the limits and headline features, there may be better deals elsewhere.

If you're still a little uncertain, InMotion's exceptional 90-day money-back guarantee gives you plenty of time to confirm the company lives up to its promises.

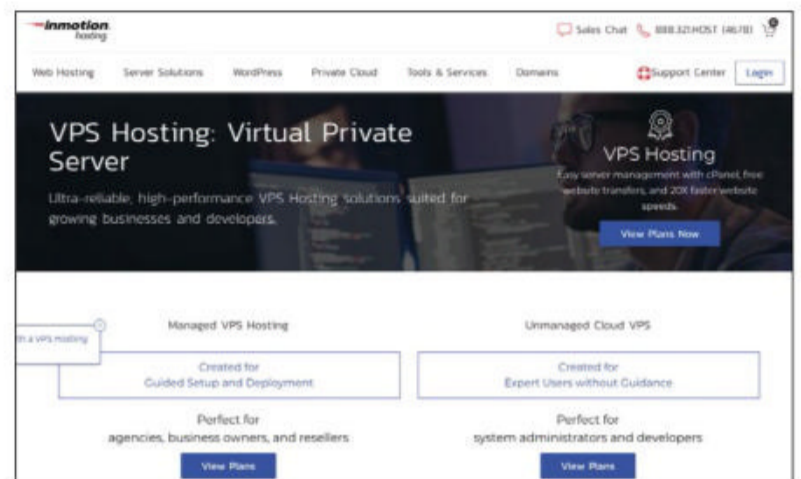
InMotion's VPS hosting plans and dedicated servers look a little more pricier than some of the competition, but they're well specified, and very good value for the functionality on offer. Managed VPS plans include free server migrations, cPanel and WHM control panels and automatic server backups, while InMotion handles low-level server management and security updates.

Prices start at \$29.99 a month over two years (\$44.99 on renewal) for 4GB RAM, 75GB storage and 4TB RAM. VPS plans include a choice of PHP or WordPress speed optimisations, enabling you to get the best possible performance for your preferred technology. Managed plans also include 'Launch Assist', an InMotion feature that gives you two hours dedicated time with a system administrator to help with any task.

InMotion's dedicated server range also gives you plenty of choice, with managed and unmanaged plans, a good range of base servers with several configuration options, and prices starting at \$99.99 a month on the annual plan. As with shared hosting, InMotion isn't the cheapest provider, but you get a lot for your money and the plans offer decent value overall.

All your questions, answered

If your hosting plan isn't working as you'd expect, the first place to look for help is InMotion's Support Center, where you'll find, well, a host of content: FAQs, product guides, tutorials, 'educational channels', a YouTube



If you're after US hosting, InMotion Hosting can handle it.

channel, web forums, and assorted web-based tools and other resources.

Drill down to an individual product area and you'll find just as much depth. For example, the WordPress Hosting section has links to more than 50 articles, covering everything from installing WordPress and logging in to the WordPress dashboard, to geek-level options like creating your own WordPress plugin.

If the website can't help, 24/7/365 support is available via email, phone and live chat. We tried a test question and found the agent was quick, helpful, took responsibility and resolved our issue without difficulty.

To assess performance we used **UpTime.com** to monitor a simple template site every five minutes from multiple locations, for a week, checking uptime and recording response times. Uptime was 100 per cent, as we'd expect. **Uptime.com** recorded two very high response times (the highest at 1.48 seconds). But that's only two samples out of more than 2,000 with an average response time of 290ms. Our most recent host tests averaged 200-400ms, putting InMotion Hosting comfortably in the middle of the pack.

We also checked site performance via Bitcatcha's server speed test and it was interesting to see our server deliver much better results, scoring Bitcatcha's highest possible A+ speed rating. **LXF**

VERDICT

DEVELOPER: InMotion

WEB: www.inmotionhosting.com

PRICE: Various

FEATURES 9/10

EASE OF USE 8/10

PERFORMANCE 7/10

VALUE 7/10

Feature-packed products and great support from an honest and professional web-hosting operation.

» **Rating 8/10**

Atari VCS

Monetising nostalgia should be an easy win for Atari so **Matt Safford** can't believe it's created another E.T.

SPECS

OS: Atari Linux OS (Debian based)
CPU: AMD Ryzen embedded R1606 2.6GHz, two cores, four threads
GPU: AMD Radeon Vega 3 1.2GHz
RAM: 8GB DDR4
SSD: 32GB eMMC fixed, M.2 slot
Comms: Wi-Fi 802.11n 2.4/5GHz, Bluetooth 4.0, Gigabit ethernet
Ports: HDMI 2.0, four USB 3.1 (two front, two rear)
Size: 29.5x15.0x4.8 cm
Weight: 1.3kg

Modelling itself after the venerable Atari 2600 that launched in 1977, the modern VCS made its debut via a YouTube teaser in 2017. The company followed that up with a crowdfunding campaign the following year, promising to deliver devices in mid-2019. Here we are a couple of years later, and the VCS has finally made its way to retail.

Available in either basic black (Onyx) or the more 2600-like Black Walnut, the device (let's call it a console, though it also has a PC Mode) sells for \$399 (£390 with P&P, VAT and Duty). You get an AMD-embedded APU, 8GB of RAM and 32GB of storage, as well as both a nostalgia-inducing joystick and a more modern Xbox-like controller. You get a promised "100 classic arcade and console games" included, as well as the ability to install popular streaming apps like Disney+ and HBO Max – in case you didn't already have a set-top box.

And in a move that at the very least makes the VCS more versatile and interesting, you can crack the console open to add additional storage via SATA M.2 as well as upgrade the RAM. A PC Mode selection in the menu enables you to install and run other operating systems, like Linux and something called a Windows.

First, the good news. As advertised, there are about 100 included Atari arcade and 2,600 games. And while we can't say we played each and every one, the titles we played worked well and looked good, certainly better than on 1980s 13-inch TV.

But then the disappointment starts. *Yar's Revenge* did not include rumble support (although some games do). And some notable titles that would require expensive licencing (like *E.T.*, *Superman* and *Pitfall!*) were absent. If you want more Atari titles, a VCS Vault Volume 2 with 50 more games is available in the store for \$5. Really though, given the price, it's irksome that the company keeps so many Atari titles behind an additional paywall. And that's nowhere near the original 400 titles.

When you get tired of retro Atari titles, the *Antstream Arcade* app also comes pre-installed, and there's real depth here. It's a retro-gaming service with "thousands" of streaming retro titles from any number of platforms



The controllers are solid and can work wirelessly with Bluetooth or wired over USB.

CREDIT: Atari

and decades, as well as online leaderboards, tournaments, and challenges. It's also free as an ad-supported service, or \$10 a month without ads.

The VCS also has a selection of additional apps and games in its store. But when you try to use the streaming apps you'll find out that you can't actually control them with the included VCS controllers. You either need to have a regular keyboard and mouse connected, or you need to use the VCS Companion app on a smartphone or tablet.

Upgradability and OS installs

Getting into the VCS is reasonably easy. The front and back panels pop off with a moderate amount of pulling and prying. The four screws that hold the VCS together are hidden under rubber feet, but the parts over the screw holes just pop out, enabling you to access the screws, and then pop right back in.

A standard BIOS means you can easily select your boot device: external USB or the internal drive. You can install your OS of choice on to either and it'll work fine. Ubuntu will install without a hitch, but the internal Wi-Fi adaptor requires its power saving to be turned off for it to work. It's perfectly capable of modern gaming, but set expectations to 720p resolutions and low settings to achieve smooth performance. **LXF**

VERDICT

DEVELOPER: Atari

WEB: <https://atarivcs.com>

PRICE: £390 Walnut bundle (£285 base unit only)

FEATURES	7/10	EASE OF USE	7/10
PERFORMANCE	5/10	VALUE	5/10

Delivers the promise of nostalgia-inducing retro gaming, but you could buy a new PC or modern console for this price.

» **Rating 6/10**



Open the Atari VCS and you can update it with extra storage!

Whonix 16

Mayank Sharma is a busy man, but when it comes to privacy and security then, like Louis Armstrong, he has all the time in the world.

IN BRIEF

The Debian-based distro is built on the concept of security by isolation. It ships in the form of two virtual machines; an arrangement that helps it create a secure environment shielded from the nasties on the internet.

SPECS

CPU: Any CPU with Intel VT-x or AMD-V
Memory: 1GB
HDD: 10GB
Build: Virtual disk images for VirtualBox and KVM.

Whonix is one of the best privacy-centric distros that does its magic using the concept of security by isolation. It comes in the form of two virtual machines (VM): Whonix-Gateway and Whonix-Workstation. The idea is to isolate the environment you work in (Workstation) from the internet access point (Gateway), which, for added privacy, routes all internet traffic through Tor.

In addition, the distro developers have done some work under the covers to deliver a hardened distro. For starters, Whonix is based on Kicksecure, which is a security-hardened Debian distro that employs several means of reinforcing the installation, such as the Linux Kernel Runtime Guard (LKRG), kernel-hardening settings as recommended by the Kernel Self Protection Project (KSPP), enforcing Strong Linux User Account Isolation, and a lot more.

The end result is a distro that guarantees IP and DNS leak protection, and ensures the user can't be identified using any mechanism in the arsenal of tricks in the online marketers' toolbox. The distro also anonymises keystrokes, and does boot clock randomisation together with secure network time synchronisation to thwart attacks based on leaked time data.

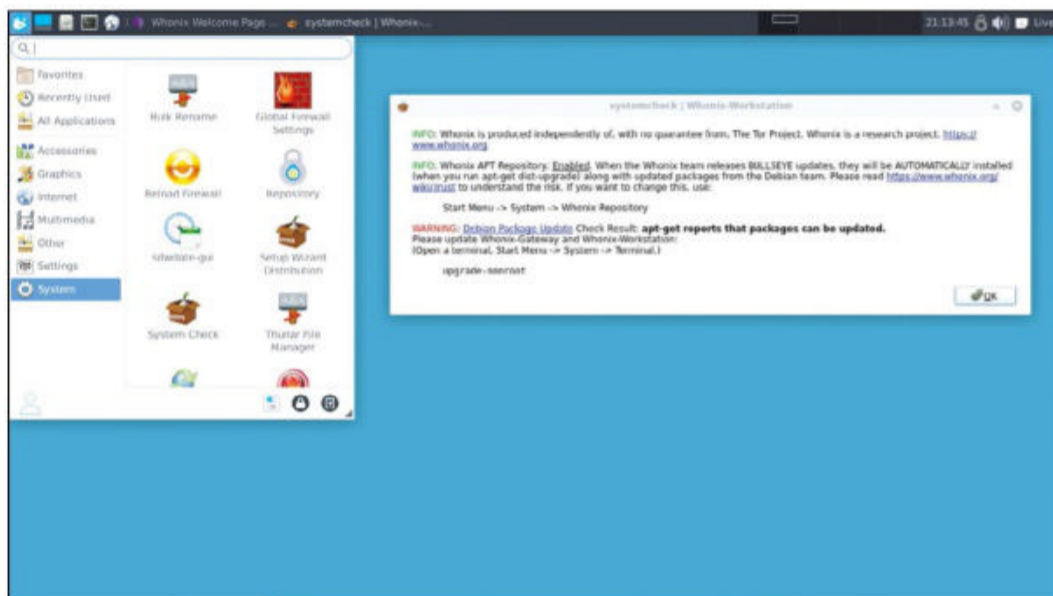
Whonix 16, the latest edition of the distro comes out two years after the project's last major release, Whonix 15, which will be deprecated in November, 2021.

The one major change in Whonix 16 is that the distro has now been rebased on the latest Debian 11 stable release. There are some minor administrative changes, such as program upgrades and tweaks to some of the repositories, but overall there's hardly any noteworthy change in the functioning or usability of the distro.

Anonymisers anonymous

Whonix continues to be available as a single appliance that deploys both VMs when imported. This also means that there's no installation mechanism for the distro, which helps make it accessible to a wide range of users.

Users are supposed to launch the Gateway before the Workstation. Since the previous release, the boot menu on both VMs also offers the option to boot into a Live environment, which prevents Whonix from committing any changes to its virtual disk. This is done to add another layer of security to the existing privacy-centric features of the distro. Both VMs use the lightweight Xfce desktop, and the Workstation VM includes a handful of programs. Most of them will work with Tor, and will protect your identity online.



Whonix ships with a handful of wrapper scripts to make regular administration tasks easier, such as changing passwords and keeping the system updated.

A primary Whonix goal is to reduce the risk posed by additional software that isn't exclusively designed to work with Tor. That said, the distro doesn't prevent users from pulling in additional software from the vast Debian repository. It doesn't include a graphical package manager, but you can use Debian's famed command-line based *apt-get* package manager. However, the distro developers do warn that this isn't something they recommend because additional software increases the possibility of attack.

In fact, Whonix, which pitches itself as a research project, is replete with documentation that will help users familiarise themselves with the nuances of desktop security. The distro is also useful for advanced users who wish to host location hidden services based on the Tor network.

All things considered, Whonix is an ideal platform for anyone who wishes to stay anonymous online, even while traversing through the notoriously bad lands of the internet. The distro isn't designed to be used as an everyday distro, since signing into any online service would defeat the whole purpose of using Whonix. **LXF**

VERDICT

DEVELOPER: Whonix Developers

WEB: www.whonix.org

LICENCE: Several

FEATURES	8/10	EASE OF USE	8/10
PERFORMANCE	7/10	DOCUMENTATION	9/10

A capable privacy-centric distro that aims to keep users anonymous and safe from being tracked or targeted online.

» **Rating 8/10**

Devuan GNU+Linux

The backlash against Systemd resulted in this sysvinit-friendly distro, now on version 4. **Mayank Sharma** assesses its progress.

IN BRIEF

A Debian-based distro that came into existence as result of Debian shifting to Systemd. Not only has it spawned a new category of distros, it's also grown into a mature project that caters to several different use cases – much like its upstream distro.

SPECS

CPU: 1GHz
Pentium class
Memory: 1GB
HDD: 10GB
Build: Available in both 32- and 64-bit variants

Debian's shift to a new system and service manager in 2017 instigated a long and much-publicised debate, which led to several Debian developers creating Devuan by swapping out the new *Systemd* in favour of the established *sysvinit*.

Devuan continues to be based on Debian, and the latest version of the distro, Devuan 4 "Chimaera" is based on the latest stable Debian 11.1 release.

One of the things Devuan inherits from Debian is the support for a wide range of hardware. In addition to being one of the few projects that still produces installation media for both 32-and 64-bit machines, its community also builds images for ARMel, ARMhf and the IBM Power platform.

Devuan's claim to fame is its support for multiple *init* systems that aren't *Systemd*. The latest release offers users a choice between *sysvinit*, *OpenRC* and *runit*. As *Systemd* expands in its scope, the Devuan developers toil to ensure that none of it seeps into the distro. For instance, the inclusion of the *udev* device management daemon into *Systemd* saw Devuan switch to *eudev*.

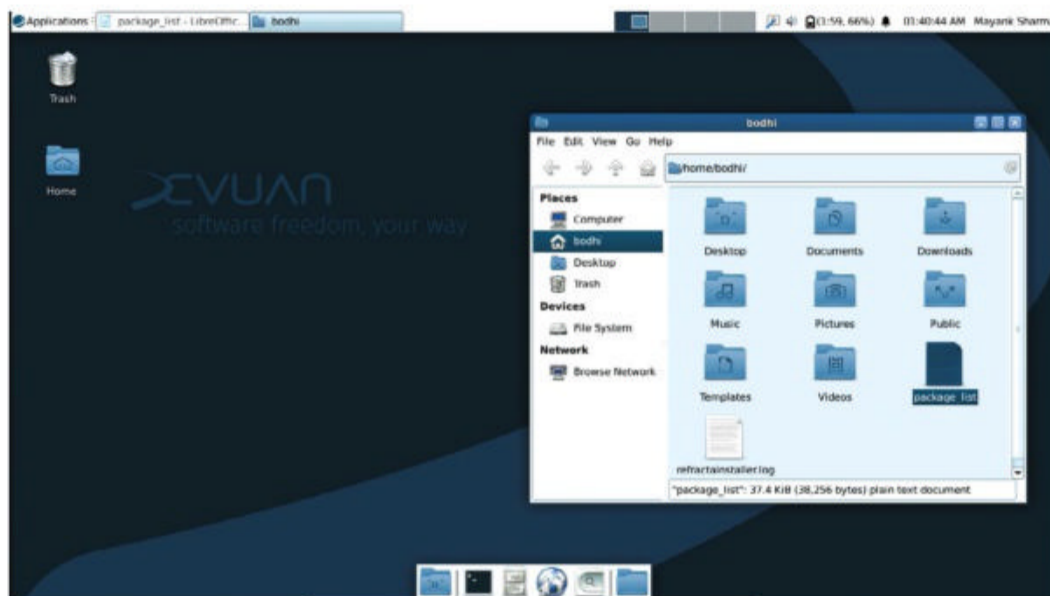
A matter of choice

Devuan produces separate installation images for desktop and servers. Advanced users are also provided with a *netinstall* installation image to tailor their installation to suit their requirements, while desktop users have the option to take the distro for a spin using the *Live* installable medium.

Talking of installers, the distro strangely uses a couple of them. The install-only media use a modified version of the standard Debian installer, which is easy to navigate. However, the Live images rely on the *Refracta* installer, which is fairly cumbersome to operate. For instance, while its plethora of options would be appreciated by experienced campaigners, they'll probably just disorient the average Linux desktop user. You can use the default options with a majority of the settings and still end up with a usable installation, but the constant interruptions quickly get irritating.

The Live CD uses the Xfce desktop environment, and that's what it'll install when you anchor the distro to your disk. On the other hand, the installer on the install-only medium offers a choice of five desktop environments including Xfce, Cinnamon, KDE, LXQt and Mate. As expected, you can install others from Devuan's repositories post-installation.

Besides the choice of desktop, you also have the option to choose from one of the three supported *init*



While creating a Systemd-free distro is no small feat, it makes for very curt release notes. The developers have thrown in a new lick of paint to visually distinguish the releases.

systems. However, neither of the installers give you the option to customise the selection of programs for your installation, although both will give you all the usual desktop productivity tools.

To help you flesh out your installation, Devuan relies on the *Synaptic* package manager and doesn't include an app store, much like its progenitor. Furthermore, while you can pull in support for Flatpak from the official repos, you can't get snaps to work on Devuan, since that particular universal packaging format relies on *Systemd*. On that note, Devuan hosts its own package repository, which mirrors the upstream Debian repositories, except for stripping away any traces of *Systemd*.

You can think of Devuan as essentially Debian without *Systemd*, but there's a lot of effort that goes into creating it. In fact, if you want to use Debian, and don't particularly need *Systemd*, we wouldn't hesitate to suggest Devuan. After all, it offers a choice of some of the leading desktop environments, support for multiple architectures, a comprehensive repository of packages, and images to cater to several use cases, from the desktop to the server. **LXF**

VERDICT

DEVELOPER: Veteran UNIX Admins

WEB: www.devuan.org

LICENCE: GPL and BSD

FEATURES **8/10**

EASE OF USE **8/10**

PERFORMANCE **8/10**

DOCUMENTATION **7/10**

Devuan is best appreciated by those who know what it takes to give users choice, right down to the *init* system.

» **Rating 8/10**

Zephix 3.0

A fan of portable, modular distros, **Mayank Sharma** runs across a relatively new one and wonders how it stacks up against his all-time favourite: Slax.

IN BRIEF

Zephix is a Debian-based, portable distro that's designed to be run from a DVD or a USB device. It describes itself as lightweight and modular, in that it ships with just the basics, and prompts users to install any additional programs as modules that can be downloaded and inserted into the distro.

SPECS

CPU: 1GHz
Memory: 3GB
HDD: N/A
Build: 64-bit only

Zephix belongs to the small minority of distros that's designed to be used from removable media, such as USB drives. In terms of features it can't match its established peers – Slax, Porteus and Puppy Linux – but it's not far from the mark, and does bring its unique touch to the field.

Like some of its peers, Zephix ships as an ISO, which you'll have to loopback mount before copying its contents on to an ext4 formatted disk, capping off the process by running a script from inside the USB drive to make it bootable.

The first signs of the distro's immaturity is its simplified boot menu, especially when compared to its peers. The distro, however, boasts the same modular nature, with just about everything available as an add-on module. The default image includes a fairly barebones graphical environment, and everything else from the web browser, and image editors, to remote desktop clients and IDEs is offered as a separate downloadable module.

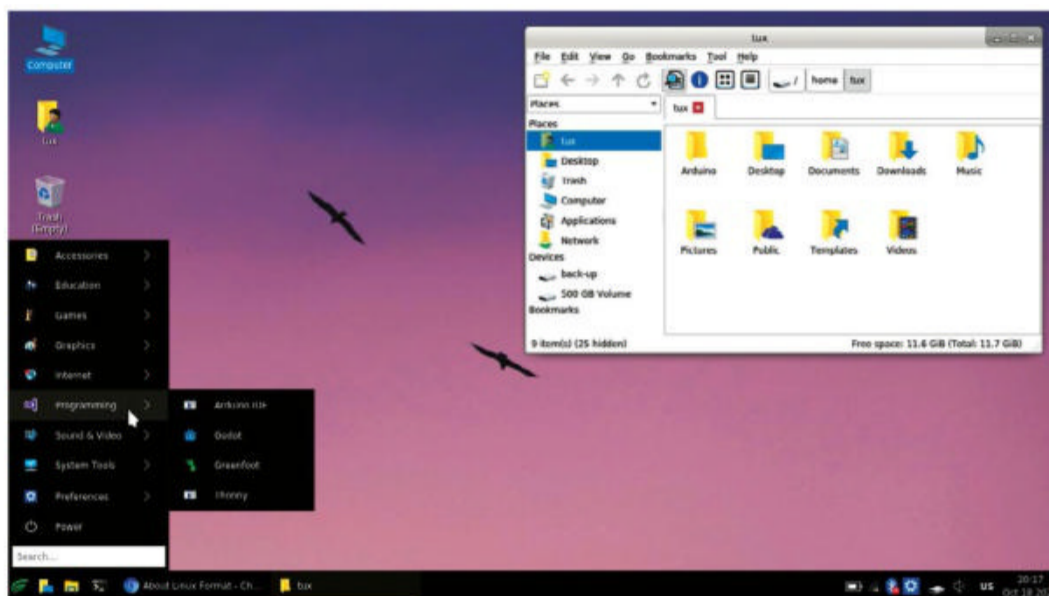
One of the unique characteristics of the distro is its customised LXQt desktop that's been tweaked to mimic Windows 10. We're slightly biased against Windows-resembling customisations, not denying that there's a huge market for Windows-draped Linux desktops. However, Zephix has done a wonderful job and its Windows lookalike desktop not only looks inviting to Windows users, but also doesn't look alien to Linux users familiar with the traditional desktop metaphor.

Cutting down on bloat

Surprisingly, customisation is both Zephix's advantage and its Achilles' heel. While the desktop customisation works in the distro's favour, we aren't pleased with the process for customising the distro, and making the changes persists across reboots. To make any of your changes available to you on the next boot, you'll have to save them as a custom module. The distro comes equipped with simple-to-use and adequately documented command-line tools for this purpose, and they work flawlessly.

What annoys us, however, is the cumbersome process. After generating the custom modules, you'll first have to move them to a writable medium, such as another plugged-in drive, or across the network. Then shut down the running Zephix installation, mount the drive under another platform, and copy over the custom module along with the other modules.

You can now reboot Zephix and have your custom files and configuration changes available to you. While it



Zephix's developer wants to make the distro even more modular in the future, trimming it down to just a CLI, and even bundle the graphical desktop as a module itself.

doesn't sound like much, doing this for saving every change can get tedious, especially when compared to Slax, which offers persistence by default.

Another minor niggles with the distro is the use of the *connman* network manager. For starters, it isn't the most intuitive network manager out there, thanks primarily to its plethora of features. Second, it isn't integrated properly into the distro. For example, we had to manually edit and point `/etc/resolv.conf` to the correct nameserver, before we could use Zephix to browse the web.

Of course these are minor niggles that mostly bugged us because of how we used the distro. If you're coming from Windows and all you need is a zippy and modular distro that offers you all your favourite tools without imposing too much of an overhead, then Zephix in its current state offers everything you need.

In fact, our only real criticism of Zephix as compared to some of its peers is the process of making changes persist across reboots. But if you don't need to save changes on a regular basis, you should definitely take Zephix for a spin. **LXF**

VERDICT

DEVELOPER: Robert Spiteri

WEB: <https://zephix.rf.gd>

LICENCE: GNU GPL

FEATURES	7/10	EASE OF USE	5/10
PERFORMANCE	9/10	DOCUMENTATION	3/10

Zephix is suitable for anyone looking for a modular distro. However, it needs to simplify its persistence mechanism.

» **Rating 6/10**

Roundup

Reiser5 » Ext4 » Btrfs » XFS » NTFS



Alexander Tolstoy

is excited to discover which filesystem is the most robust and reliable of all.

Filesystems

It's time for **Alexander Tolstoy** to put the major Linux filesystems to the test and figure out which is best in terms of speed, reliability and extra features.

HOW WE TESTED...

We gave extra attention to performance in this *Roundup* and conducted a series of tests to measure the filesystems' capabilities. The hardware setup was built to handle four types of storage under a single Linux workstation. We went with the USB 3.1 Transcend Jetflash stick, Seagate Exos 7E2 hard drive (7,200rpm/128MB cache), Intel 530 Series SSD and Samsung SSD 970 EVO Plus NVMe drive. We decided to conduct a synthetic test for measuring read speed using *Postmark*, which does a good job in stress-testing I/O using a large number of transactions, giving us figures for sequential reads. As for writing speed, we engaged *Sysbench* with its fileio mode to measure synchronous write rate on a series of 2GB files. Lastly, we tested the unpacking of the Linux kernel tarball, which was about 120MB and consisted of nearly 60,000 items – a typical example of real-world file operations. We also conducted a power outage imitation test to see how our contenders handle not being properly unmounted.



This is not the first time we've taken a close look at filesystems here in *Linux Format*, and you may recall our previous *Roundup* in **LXF203**. That was over six years ago now, a lot has changed since then. Btrfs, in particular, was markedly less mature over half a decade ago, especially when you consider the fact that it has now been adopted as a default filesystem for root partitions in mainstream Linux distributions, such as Fedora and openSUSE. We wanted to see if it's as fast as Ext4 (our previous winner) and also check whether Btrfs has any downsides that could be detrimental to its usage on Linux workstations.

There's some unexpected news in the emergence of Reiser5 (reviewed in **LXF260**), a one-man project bravely led by Edward Shishkin as a next-gen replacement for the ageing Reiser4. The latter yielded very mixed results in our previous filesystem *Roundup*, but we decided to give the new experimental Reiser5 a chance and took it for a ride alongside the other filesystems to see whether things have improved. XFS, the popular alternative to Ext4, is also included for reference, while the fifth contender is the now-kernel-supported NTFS – we still have to use it from time to time to maintain compatibility with the Windows world.

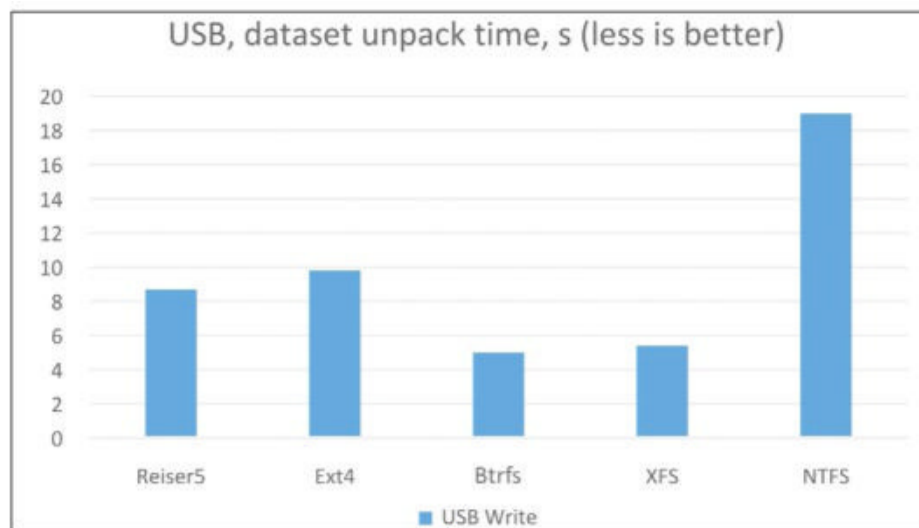
USB flash drive performance

What if your USB thumb drive could perform faster?

We took a typical USB 3.1 thumb drive and performed three types of test. One was *Sysbench* with its fileio mode, which we used for synchronous reads and writes. It's a synthetic test, but it does stress a filesystem far more than most manual file operations. Second was the *Postmark* read test (100,000 transactions), while the third was the dataset test, which involved unpacking a Linux kernel tarball on to the drive.

The results turned out to be interesting and varied between *Sysbench*, *Postmark* and the dataset unpacking. The more strenuous *Sysbench* test revealed that Ext4 was the fastest filesystem in terms of simultaneous read/write operations that involved large files already on the drive. However, when it came to a more dense flow of thousands of tiny files, XFS was the best according to both *Postmark* and the dataset test. Btrfs seemed to reside in the mid-range, with fairly good results, although we doubt whether there's any sense in using the snapshot feature on flash storage – but who knows? Btrfs showed very balanced results anyway. Reiser5 ran rather well in all tests, even outperforming Ext4 in the dataset unpack test.

The NTFS results were expectably poor due to the fact that so far we have only got a slower FUSE-based NTFS-3G driver, which often sees huge slowdowns, though with some exceptions (for example, *Sysbench* seems to favour NTFS). The downsides of NTFS-3G dissolve when we work with large files on our USB drive. That said, in certain scenarios, you won't notice any difference



Use something other than NTFS-3G on a flash drive; whatever you pick, you'll see swifter performance. As this chart shows, Btrfs and XFS are the fastest filesystems.

between NTFS, XFS or Reiser5 on a flash drive. Summarising all observations, XFS seems to be the best option for a flash drive when dealing with small operations, while Ext4 is better for more serious I/O. Btrfs is also very strong, but you need to be cautious and keep an eye on things such as available free space (Btrfs health greatly depends on it). The safe option remains Ext4, which topped the chart in *Sysbench*, but lagged a little in the dataset test. It's still a very fast and reliable filesystem, which works well with various types of storage, flash being no exception. XFS shines when dealing with small files, but loses its glory on heavier loads, which shows very sharply when stressing a USB flash drive.

VERDICT

REISER5	6/10	XFS	7/10
EXT4	8/10	NTFS	4/10
BTRFS	9/10		

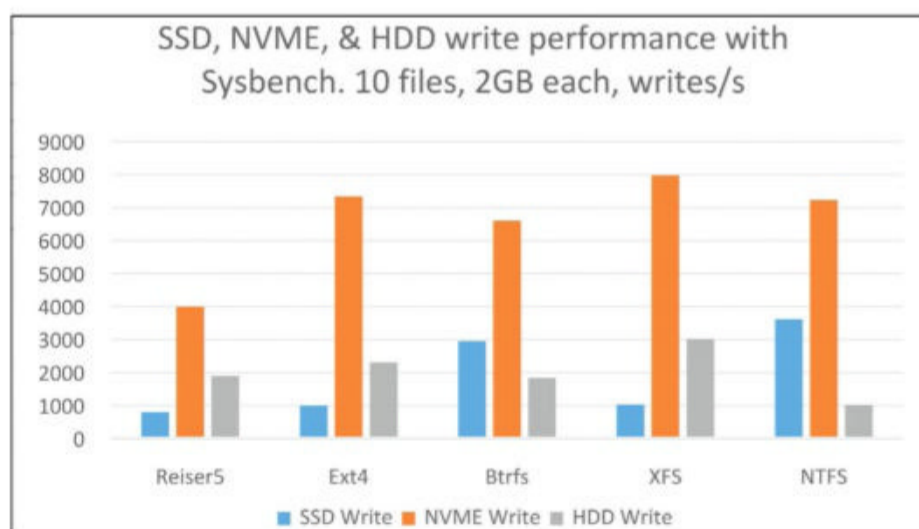
If you need to squeeze more speed out of a USB thumb drive, try formatting it with the Ext4 or Btrfs filesystem.

Solid-state drive performance

It's the most widespread drive type in modern personal computers.

We applied our testing technique to the Intel 530 Series SSD. Looking at the performance charts, there's only a subtle difference in how Ext4, Btrfs and XFS perform. You could pick any of those and get the same performance; they're all very good. If you need a reason to choose one filesystem over another, read and write speeds won't suffice, so consider extra features (see *overleaf*). Reiser5 showed a lacklustre performance during the read test, though it was still decent – at least, it was better than the old Reiser4 figures. Nonetheless, Reiser5 ran close to the winners, just 10 per cent slower in the dataset test, and about 30 per cent slower in *Postmark*.

NTFS showed very good results in *Sysbench* but otherwise it was poor and couldn't perform anywhere close to its competitors. In *Postmark* it was six times slower for reads, and 2.5 times slower for writes. We first thought that the NTFS-3G driver from Tuxera was to blame. It's known to be a user-space driver that works with the FUSE subsystem in Linux and not normally about speed. Still,



Sysbench tried to convince us that NTFS was fast, but it was the effect of writing existing large files. In real-world use, you would see abysmal performance in Linux.

that's no excuse because most Linux users don't have anything else to choose from and are forced to deal with NTFS-3G. Paragon's much faster NTFS driver will soon land in the Linux kernel, but until then we're stuck with that slowness.

VERDICT

REISER5	7/10	XFS	9/10
EXT4	9/10	NTFS	4/10
BTRFS	10/10		

On convenient SSD drives, Btrfs performs better than the rest. Also consider Reiser5 – it works very well on SSDs.

Extra features

Weigh up a range of other capabilities.

Although we undoubtedly value speed over pretty much everything else when looking at the merits of any particular filesystem, sometimes that isn't enough to differentiate any one over the rest. In that situation, we need to take other things into consideration. For instance, you might be choosing a filesystem for a more complex setup, such as a RAID array or an LVM volume, in which case you need to investigate each filesystem's more advanced features. On top of this advanced extras such as automatic checksums and snapshot options all help secure your files and provide additional security. Let's take a closer look at our five contenders to find out whether they have anything else to offer.

Reiser5

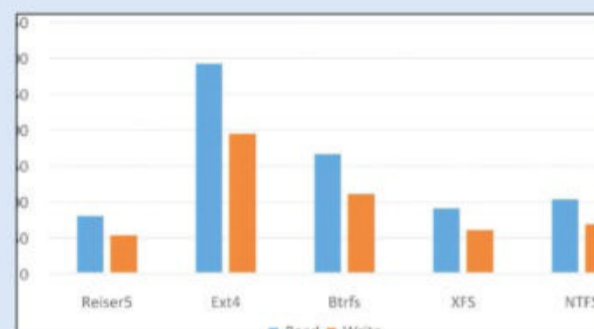
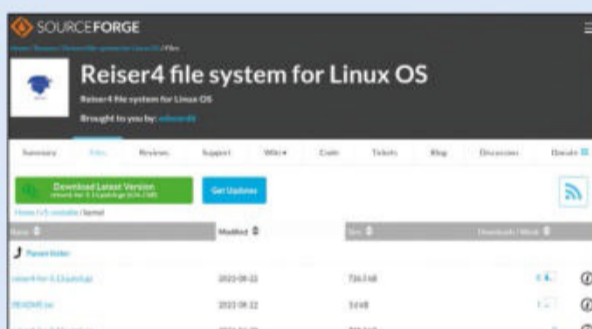
8/10

Ext4

3/10

About a year ago, Edward Shishkin – the sole Reiser5 developer – reported that he had stabilised the filesystem's logical volume functionality. So, the real power of Reiser5 doesn't lie in formatting a single plain drive partition. Instead, it's found in enterprise-grade usage, where several storage devices are united in one logical device. The logical volume manager in Reiser5 supports an impressive feature known as Data Tiering, which involves dumping peaks of I/O load to a proxy device. The idea is to add a speedy drive to an existing array of slow drives and make the whole array perform quickly. As long as I/O peaks don't last for too long, the Reiser5 volume balancer has enough time to flush the cache and make the fast device available for the next peak. The ability to balance data between parts of the volume and fast data migration make Reiser5 very promising.

There's no urgent need to tweak your existing Ext4 mounting options because your system is already using the best defaults. So, in this regard, there's not much to explore here. Ext4 is rock solid, very fast (although it tends to lose out to Btrfs for first place) and reliable. Ext4 rarely fails, and when it does, chances are that the real source of the problem is the drive itself. Ext4 also benefits from a lower fragmentation rate than most other filesystems, thanks to delayed block allocation. That means that, even in the long run, an Ext4 partition won't show signs of slowdown, even in the case of intensive reading and writing. However, as this part of our *Roundup* is dedicated to extra features, which Ext4 lacks, we've given it a low score here. When building an advanced storage solution, you might want to consider other filesystems, which may turn out to be more beneficial.



NVMe drive performance

Expecting the same results as for SSD? Then you're mistaken...

You might think a separate test for NVMe drives would be superfluous, but it's not (otherwise we wouldn't have bothered with it). Our five filesystems displayed quite different performance on the blazingly fast Samsung EVO 970 drive with non-volatile memory, so let's see exactly how it differed from the legacy SSD drive performance.

First of all, Ext4 and XFS yielded the best results, while Btrfs dropped by one-third and looked more like Reiser5 in terms of speed in the *Postmark* test. We checked that deviation using `dd`, which enabled us to observe the reading speed of a large file, but it only confirmed our initial findings. Notice that this time all three tests indicated that Btrfs did not perform as well on an NVMe drive as it did on SSD. That said, Btrfs on the root partition of an NVMe drive will eventually result in slower application startup times and should probably not be used that way.

Another difference from the SSD performance is that the much faster nature of NVMe effectively smoothed out the differences between Btrfs, EXT4 and XFS. That means that on

smaller I/O tasks (such as our test archive untar routine), there's virtually no sense in considering one filesystem over any other – they all deliver practically the same performance. Reiser5 gained entry to the speedy folks club when it came to data unpacking, but it lagged behind by around 30 to 40 per cent in the two other tests. Nevertheless, Reiser5 showcased a decent daily driver level of performance on NVMe, and that's definitely good news. By the way, NTFS is still a big no in *Postmark*, but a firm yes in *Sysbench*. It's just a matter of the type of things you do with an NTFS partitions in Linux that makes it feel either sluggish or snappy.

VERDICT

REISER5	8/10	XFS	10/10
EXT4	10/10	NTFS	6/10
BTRFS	8/10		

We recommend using either Ext4 or XFS for those speedy NVMe storage devices. Btrfs may be a little slower than you'd expect.

Btrfs**10/10****XFS****5/10****NTFS****2/10**

Btrfs has an abundance of extra features that put it in the top spot. First of all, there are snapshots – readable and writable copies of the entire Btrfs subvolumes. You can use such snapshots as backup copies of your system, which is a really useful feature. Snapshots can't help in the case of drive failure, but they do prove useful if you need to roll your system back after a bad update, or in order to restore a missing file. Unless you create a large Btrfs pool and rotate its snapshots too frequently, your performance won't degrade. However, Btrfs does hog your CPU more than other filesystems, and it also devours the effective free space on the drive. In return, it gives you transparent compression (two copies of the same file occupy the space of one), plus a few more advanced features, such as its built-in multidisk RAID support. For this it's in well-deserved first place.

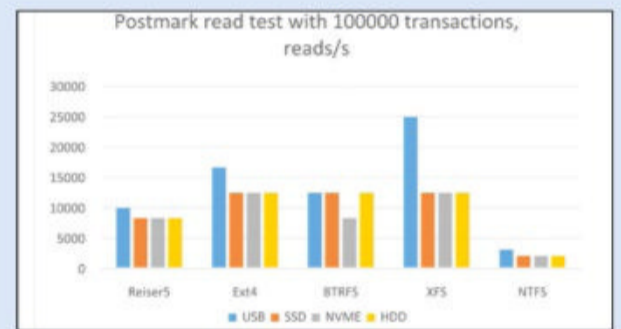
XFS has its roots in server storage, where it was used in large arrays. Before 2011, it also supported larger drive configurations than Ext4: 500TB versus 16TB. However, these days that limitation is no longer valid as the *resizefs* program that serves Ext4 partitions uses the 64-bit flag by default, enabling it to span up to 1,024 PiB. XFS supports online defragmentation and resizing of the mounted filesystem. It also claims to have sophisticated metadata read-ahead algorithms, extensive run-time metadata consistency checking, and robust repair utilities. From 2014, XFS applied the newer on-disk format (v5), which includes a metadata checksum scheme called Self-Describing Metadata, making it more better reliable. That's great for production environments but not overly impressive. Therefore, there's no explicit advantage over Ext4 and no extra features compared to Btrfs or Reiser5.

While the NTFS-3G code seems to be in good shape, the overall performance of this filesystem in Linux is abysmal, and our tests show it. The FUSE-based NTFS-3G driver can't win any performance test, regardless of the type of load. It looks as though it has no chance here either, as NTFS-3G has effectively no advanced features. The driver has been advertised as a high-quality solution for reading and writing data on NTFS partitions. Indeed, it doesn't corrupt files and if a Windows partition has left NTFS in an unsafe state (for example, Windows wasn't resumed after suspension), the driver refuses to mount it. You can happily use the current NTFS driver in Linux to check existing Windows NTFS partitions and safely write data to them, but that's about it. Paragon's forthcoming in-kernel NTFS3 driver should be much faster, but we don't expect any extra features there, either.

```

1 #!/bin/bash --
2 #
3 #
4 #
5 #
6 #
7 #
8 #
9 #
10 #
11 #
12 #
13 #
14 #
15 #
16 #
17 #
18 #
19 #
20 #
21 #
22 #
23 #
24 #
25 #
26 #
27 #
28 #
29 #
30 #
31 #
32 #
33 #
34 #
35 #
36 #
37 #
38 #
39 #
40 #
41 #
42 #
43 #
44 #
45 #
46 #
47 #
48 #
49 #
50 #
51 #
52 #
53 #
54 #
55 #
56 #
57 #
58 #
59 #
60 #
61 #
62 #
63 #
64 #
65 #
66 #
67 #
68 #
69 #
70 #
71 #
72 #
73 #
74 #
75 #
76 #
77 #
78 #
79 #
80 #
81 #
82 #
83 #
84 #
85 #
86 #
87 #
88 #
89 #
90 #
91 #
92 #
93 #
94 #
95 #
96 #
97 #
98 #
99 #
100 #

```



Spinning rust drive performance

They need extra time to spin up and are noisy, but they are still in charge.

The picture for filesystem performance on mechanical/rotational hard drives is different once again. These are the slowest type of consumer storage devices, and we expected our tests here to highlight any potential design weaknesses of our filesystems. And we were right. During both the *Postmark* test and the dataset unpacking test, we had Ext4, Btrfs and XFS performing equally well. In *Postmark*, we saw Reiser5 lagging behind by around 30%, while NTFS looked downright ugly. But when it came to dataset unpacking performance, there was a different result: Reiser5 dropped to its worst figures and performed just as poorly as NTFS. It looks as though the next-gen design crafted by Mr Shishkin is having trouble handling small I/O on mechanical hard drives. Although it looked much better on the SSD drive, this time Reiser5 fell into a pit – it was horribly slow. If you happen to use a rotational HDD, consider choosing another filesystem from our usual suggestions: Ext4, Btrfs or XFS. Any of those three options would do a good job on a mechanical drive thanks to their low fragmentation and

efficient block allocation. We tend to trust *Sysbench*, which indicated that in the long run XFS is a slightly better choice for traditional HDDs. But even a multi-snapshot setup using Btrfs on such a drive is good idea, because that filesystem can auto-defrag itself (it's a mount option). Meanwhile, the only reason to use NTFS on such a drive is compatibility – if you frequently need to move files between a Linux and Windows machine, for example. That's the main reason why people keep using this lethargic filesystem. Hopefully, things will change when Linux kernel 5.15 starts being adopted by mainstream Linux distros.

VERDICT

REISER5	6/10	XFS	10/10
EXT4	9/10	NTFS	3/10
BTRFS	9/10		

Although we've been very benevolent towards Reiser5, it dropped a clanger. XFS is our favourite here.

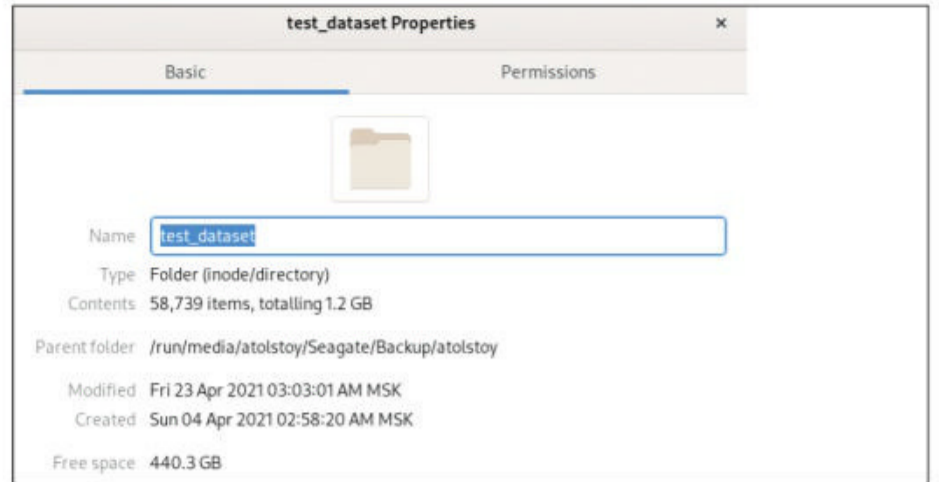
Power loss tolerance

We cruelly unplugged our external SSD while writing data to it.

This test reflects our limited experience of imitating a power outage, to see how a filesystem can handle it. Of course, we should have conducted more detailed exercises with storage devices handling high I/O with solid uptime figures. Still, despite the fact that we only got a rough impression of the filesystems' reliability, the test method was highly practical.

For each of the five filesystems, we started copying a 7GB dataset on to the external SATA-III SSD and at some point unplugged it, reconnected and checked whether we could access the already copied data. XFS, Ext4 and NTFS handled the situation very well. After several drive disconnections for each filesystem, there were eventually errors caused by the metadata not being written on the drive, but that was expected. Each of these three filesystems has repair utilities that fixed the errors.

The Btrfs story was less ideal. This filesystem is known to be tolerant of failures, and that was indeed true, until we finally had a completely broken partition after reconnecting the drive. If you encounter this, don't use the `btrfs check --repair` command, which will fix the partition, but completely erase all the data on it.



The Linux kernel tarball turned out to be a perfect dataset for measuring writing speeds. We dropped the system cache before running `untar`, and used `sync` after.

As for Reiser5, its performance was terrible in this test. Unplugging a Reiser5 drive without proper unmounting rendered its partition blank. Furthermore, in four out of five attempts, the whole host Linux system locked up – presumably due to the reiser4 kernel module not properly handling the loss of the target block device. Take care!

VERDICT

REISER5	3/10	XFS	10/10
EXT4	10/10	NTFS	10/10
BTRFS	5/10		

The NTFS-3G driver may be slow, but it won't corrupt your data. That's not the case with Reiser5, which turned out to be very fragile.

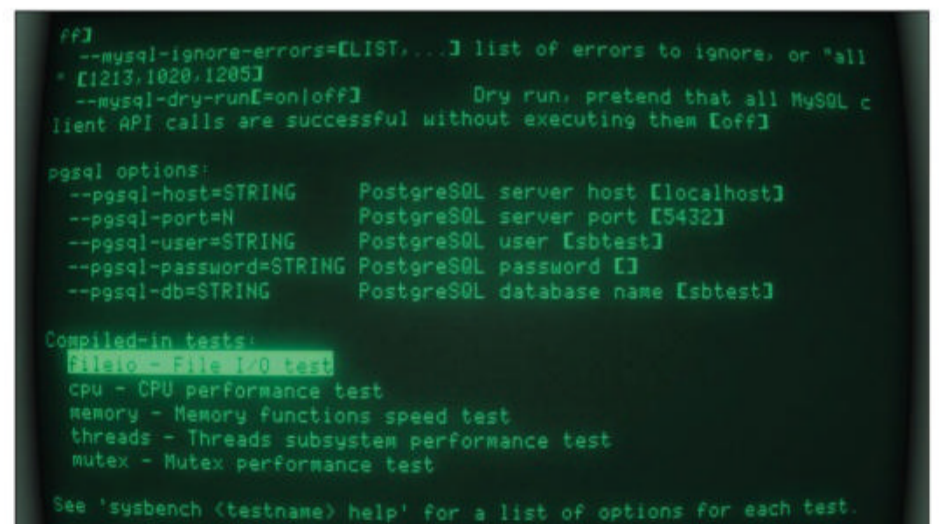
Ease of use

What does it take to set up a filesystem and use all its features?

There are virtually no extra steps required to use Btrfs, Ext4 or XFS. For years, they've been included in the Linux kernel, which means that any Linux system can mount such partitions without installing or configuring anything else. Formatting, growing, fixing filesystem errors and similar service tasks are performed by command-line utilities found in the `btrfs-progs`, `e2fsprogs` and `xfsprogs` packages available in any Linux distro. The same goes for NTFS – most mainstream distributions are normally capable of reading and writing on NTFS partitions.

So far so good, but we need to subtract a few points from Btrfs. Despite Btrfs's killer feature being the built-in snapshots, you need to look for a tool to manage them. Some third-party tools do exist, but there isn't any industry standard desktop integration for Btrfs snapshots similar to the time machine-like slider in the tweaked `Nautilus` version in OpenSolaris. We're left with the CLI toolset from `btrfs-progs`, which is fine, but it sets a higher barrier to entry for novice Linux users. If you really want to go hardcore, though, try Reiser5. This filesystem is the most difficult to get up and running. For various reasons, the code isn't part of the mainline Linux kernel, thus you need to manually get the kernel sources, patch it with the correct version of the Reiser5 patch, rebuild the whole thing, and then separately build the `reiser4progs` toolset.

There's one more confusing thing: there is the older reiser4 kernel code and the newer one with the same name (but in



sysbench comes with different benchmarks, including the file I/O stress test mode, which was perfect for the sequential write test we wanted to conduct.

another branch). Consequently, there are two versions of `reiser4progs`, plus the extra kernel feature for making the new filesystem format compatible with the previous one. You need to be very passionate about Reiser4/5 to get it right. There's an easier way, though: the Metzli project (<http://metzli.it>) produces a custom Debian 11 minimal ISO, which comes with all things Reiser5 and uses that filesystem for its root partition. Perhaps this is the most unintrusive way of giving Reiser5 a whirl.

VERDICT

REISER5	1/10	XFS	10/10
EXT4	10/10	NTFS	10/10
BTRFS	8/10		

Rotating Btrfs snapshots means you need to manage your filesystem from the command line. Old-world filesystems such as Ext4 and XFS are simpler.

The Verdict

Filesystems

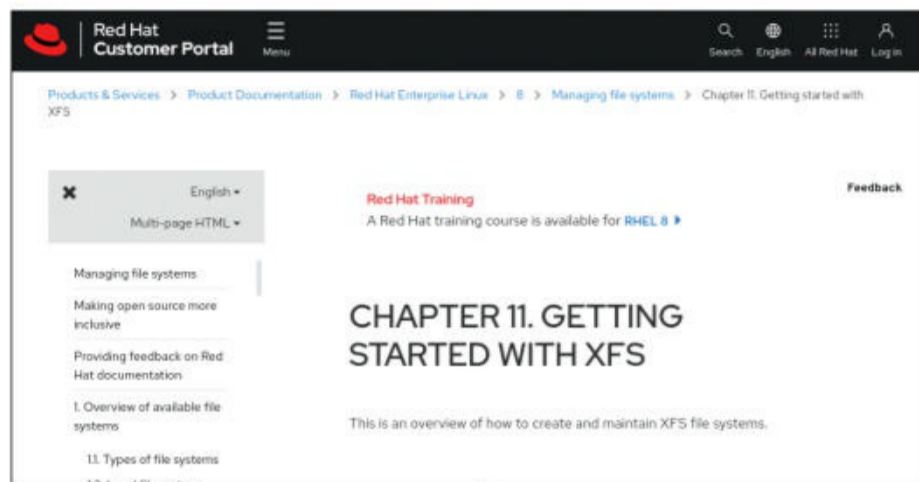
And the winner is... XFS! It scored the most points in our tests and made the best impression. XFS is super-fast, rock solid and universal. You don't need to be aware of its design or learn about the stripe-aware policies or delayed space allocation fused inside XFS. Just format your drive with XFS and you have nothing to worry about, whatever type of drive you use: mechanical, SSD, NVMe or USB flash. Nothing's perfect, though, so we still recommend keeping a backup of your data elsewhere. We couldn't manage to break XFS and deliberately lose data, but you may not be so lucky. The advantage of XFS is its handling of lots of small files on USB flash storage. Frequent, incremental writing and random reading from flash storage are where XFS shines, delivering excellent performance.

Next is Ext4, the best-known and most widely used filesystem on Linux. Apart from the USB performance test, Ext4 is as good as XFS. Historically, most experts recommended using Ext4 for the root partition (`/`) and XFS for `/home`, and we see no reason why this advice isn't still valid.

In third place is Btrfs. This filesystem has been under active development for the past decade. It shows very good performance on most types of drives, but brings some overheads. How can it maintain its live incremental snapshots without compromising speed? The answer is in higher CPU usage and the fact that you can run out of free disk space even if you think you have enough. Btrfs has been promoted as a default root filesystem in several Linux distributions – Fedora and openSUSE to name just two – but we're still not sure whether it can survive a sudden power outage as well as Ext4 or XFS.

Next is Reiser5. We need to stress that the latest Reiser5 is far faster than the old Reiser4 in most cases. Despite the fact that we didn't use logical volumes within Reiser5, its regular performance was good. Looking at our synthetic and real-world usage tests, it would be fair to say that Reiser5 is roughly 15 per cent slower than Ext4 in regular reading/writing operations when used on everything except for mechanical hard drives (sadly, it's simply too slow there). And, let's not forget that Reiser5 was easy to break, so make sure you always unmount it properly if you want your data to remain uncorrupted.

As for NTFS-3G, it's rock solid yet painfully slow. The newer NTFS3 driver should fix it – and early tests report it to be five times faster than NTFS-3G. But even if it was already that fast, it wouldn't help NTFS much to a win in this *Roundup*.



1st **XFS** **10/10**

Web: <https://xfs.wiki.kernel.org> **Licence:** GPL

Version: –

Its power is in the absence of any notable shortcomings.

2nd **Ext4** **10/10**

Web: <https://ext4.wiki.kernel.org> **Licence:** GPL

Version: –

A super-fast and reliable filesystem, although a bit slow on flash drives.

3rd **Btrfs** **8/10**

Web: <https://btrfs.wiki.kernel.org> **Licence:** GPL

Version: –

Very robust, with lots of features, but we doubt it has enough failure tolerance.

4th **Reiser5** **6/10**

Web: <https://sourceforge.net/projects/reiser4/files/v5-unstable/> **Licence:** GPL **Version:** 5

Amazing features for logical volumes, yet it's very unreliable for daily use.

5th **NTFS** **4/10**

Web: <https://github.com/tuxera/ntfs-3g> **Licence:** LGPLv2 **Version:** 2021.8.22

Very solid and production ready, but also shows abysmal speed rates.

» ALSO CONSIDER

Of course, there are many other filesystems available for Linux that you might consider useful. We're looking forward to testing the new Linux kernel 5.15 with the faster NTFS3 driver. You're not supposed to use it for a root partition anyway, but we wonder how robust Paragon's new NTFS3 driver is going to be.

We also haven't touched upon filesystem options such as JFS, F2FS, GlusterFS or ZFS, as well as the legacy stack presented by EXT2/EXT3 and the historic reiserfs 3. Some

legacy systems and custom-made hardware configurations can still benefit from using one of those filesystems, be it for reasons of compatibility, speed or general reliability.

When choosing a filesystem, feel free to conduct your own tests according to your personal usage scenario. Different combinations of storage hardware with a filesystem can sometimes yield unexpected results, so it could be a question of trial and error. **LXF**

UBUNTU VS FEDORA

Jonni Bidwell wants to know everything – and he means *everything* – about the two most popular Gnome-based distros...

Fedora and Ubuntu are both highly regarded distros, but have different approaches in a number of areas. If you were to believe the first few Google results comparing them, you'd conclude that Ubuntu is more suited to beginners, that Fedora features new technology first, and that both have large companies backing them. But these listicle summaries rarely tell the full story, so to celebrate the release of new versions of each we thought it'd be a fine time to really put these OSes to the test.

We'll look at software availability, gaming prowess as well as some technical points about how each is put together. The flagship releases of both distros run Gnome and both use the Wayland desktop protocol, so there's not much to compare there. The interim Ubuntu releases are supported for nine months, whereas Fedora is supported for only seven. If these two months matter to you, you've already got some use out of this feature. But if you want to know more about how in-place upgrades work for both, then you'll have to read a little further.

We'll also look at each specimen's server offerings. Ubuntu's *cloud-init* tool makes it easy to set up a new server, and Fedora's *Cockpit* tool will have you administrating like a pro in no time. If you're into IoT then Ubuntu Core with its Snap-powered modularity will get your embedded projects up and running. Fedora's CoreOS Linux is ideal if you want to run container-based workloads. And there's also Fedora Silverblue, powered by OSTree atomic updates.

Okay, time to pit Orange against Blue in a fight to the, urm, kernel panic.

What makes them great, again?

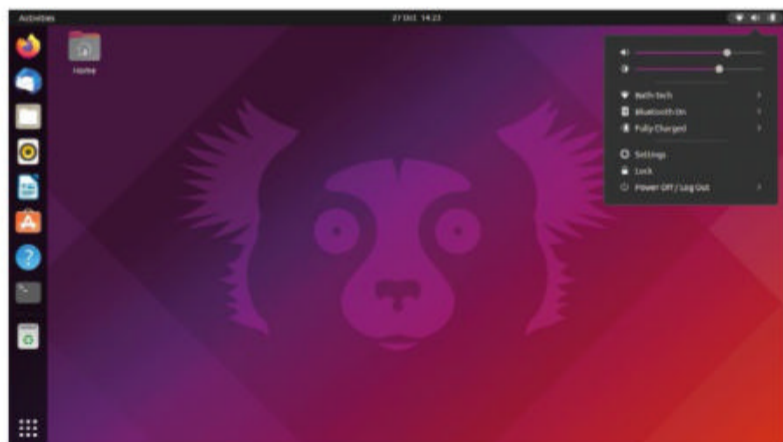
Here at Linux Format Towers we're always recommending both Ubuntu and Fedora, but sometimes we forget why...

There's a school of thought that states Linux is also all about choice. Then again, there's also the website <http://islinuxaboutchoice.com> which says different (and in very large blue letters, too). Hearsay and single-page websites notwithstanding, users certainly do have a choice about which Linux distribution to use. And sometimes that choice is difficult.

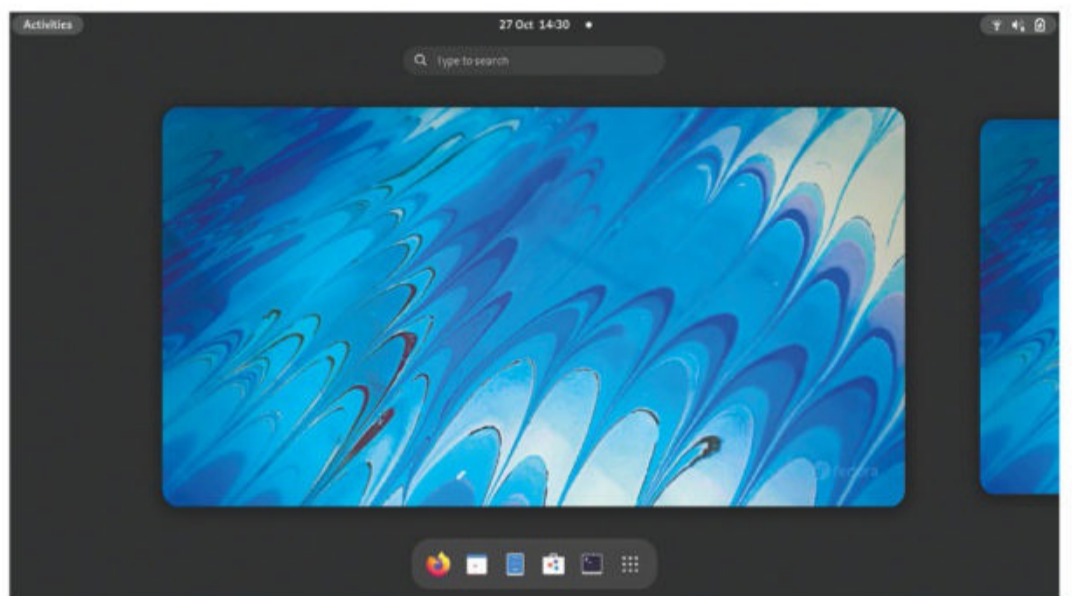
Ubuntu is often classed (along with its derivatives Mint, Pop!_OS, elementary OS and Zorin OS) as a beginner-friendly distro. Fedora, by comparison, is seen as a testbed for new (and especially Gnome-related) technologies that's more suited to intermediate users. But this definition isn't entirely fair. A beginner (with just a little bit of luck and no Nvidia hardware) would probably get on just fine with Fedora. And if they don't then it's unlikely they'd fare much better with Ubuntu, where the only obvious user-facing difference – an Ubuntu-themed dock on the left-hand side – is unlikely to provide any kind of moral support.

Stepping up a gear

Advanced users revel in both operating systems, too. The security-conscious among them approve that



The Impish Indri is already one of our favourite Ubuntu animals. It's a little taxing not being able to switch audio devices from the controls.



AppArmor (Ubuntu) and SELinux (Fedora) offer incredible granularity for locking down applications. They like the harmony that goes with having the same software stack on desktops and servers. Ubuntu gives users with exabyte storage requirements (or just people who like advanced filesystems) an experimental option to install on ZFS. Fedora now uses Btrfs (the Btree filesystem, annoyingly referred to as 'butterfs' by fans of dairy products) by default, which can likewise cope with data spread (*butter?–Ed*) across multiple huge drives.

Thanks to Snaps even users of the Ubuntu LTS can get hold of bleeding-edge software in a single click. Those seeking newer kernels and low-level system tools (only available as traditional RPM and DEB packages) will find them in Fedora and the interim Ubuntu releases, which is what we're going to focus on in this sequel. Snaps are perhaps a little more versatile than Flatpaks, because they can package command line utilities as well as graphical applications, but both offer potentially increased security through sandboxing and isolation features. And both are much more convenient than fiddling around with third-party repositories.

Fedora introduces you to the new lateral workspace arrangement by logging you straight into the Activities view.



» FREE AS IN “WHY WON'T MY FREEKING WIRELESS WORK?”

Apropos to our licencing feature (wow our technical editor is good), Ubuntu and Fedora have fairly divergent policies as to what can and what cannot live in each distributions repositories. We mentioned ZFS earlier, which has its origins in Sun's Solaris operating system. It was open sourced in 2003, but done so under the Common Development and Distribution License (CDDL). As such it can't be part of the Linux Kernel proper, since CDDL code can't be relicenced under GPLv2.

But the ZFS on Linux (ZoL) project has engineered a kernel module that Canonical is apparently happy enough to distribute with its OS. Fedora doesn't have any truck with non-free offerings, and as such ZoL, the proprietary Nvidia driver and various bits of Wi-Fi firmware all require remedial steps to install there.

This stance on software freedom shouldn't necessarily be a reason to not use Fedora. It's easy for owners of Nvidia hardware to get Fedora to be set up for

AAA-gaming. Just add the **RPMFusion-Nvidia** repository in the same way as (over the page) we add the non-free repository for the Intel Media SDK.

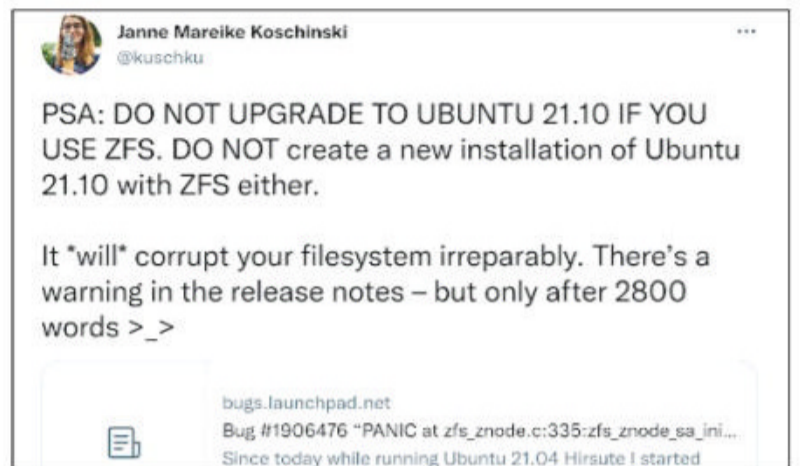
Fedora generally gets new technology working before other distros, with the exception of bleeding-edge ones like Arch and Gentoo. However, in those distros even though the new technology is there, getting it to actually work without breaking something else is often a challenge (*that we enjoy, right? – Ed*).

What's in a top-tier distro?

Learn how Fedora and Ubuntu are engineered, governed and supported.

Just as anyone can contribute to the Linux Kernel, so anyone can contribute to Fedora or Ubuntu. You don't need to be a seasoned coder – there are always translation and documentation tasks to do. If you're a dab hand with a (virtual) paintbrush maybe you could contribute some icons, themes and logos too. Distro development isn't some communist free for all, though. There are committees and managerial structures, though these are in general much less rigid than you'd find in a similar-sized company. In 2015 some 35 per cent of the 2,000-odd Fedora contributors were Red Hat employees, though the remaining 65 per cent may well have been working for someone else.

We won't get into the technical minutia of how a distro is actually made. Look at the appropriate `-next` branch of any distro's GitHub and you can get an idea of the process. For illustration though, a week after the release of Ubuntu 21.10, the first daily builds of 22.04



Bugs are inevitable, but ones that corrupt data are the worse. This one in ZFS got fixed while we wrote this feature. CREDIT: Twitter.com

OpenSSL 3 and Ruby 3.0 are both slated for release in November, for example. If you're upset that Ubuntu 21.10 missed out on Gnome 41 (by dint of misaligned release schedules), you'll be pleased to hear that Gnome 42 will (assuming no delays to its release) be powering the Ubuntu 22.04 desktop.

THE ART OF GOOD REPORTING

“If you think you've found a bug then familiarise yourself with the bug-reporting process for each distro, so that the developers can fix it”

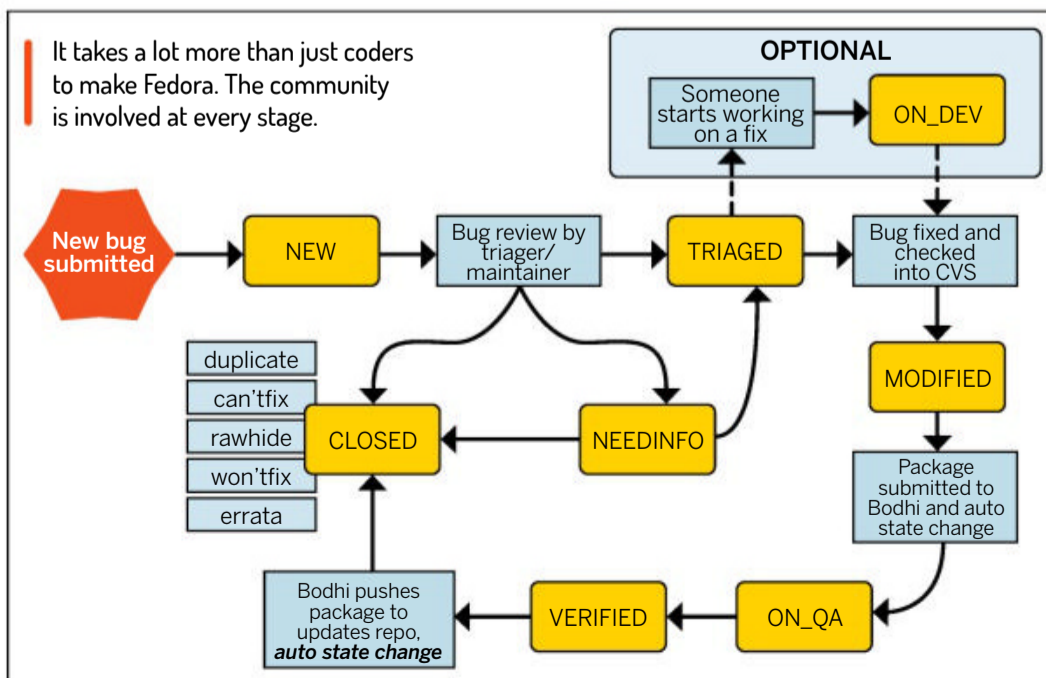
(Jammy Jellyfish) appeared. These at the time of writing hardly differ at all from the 21.10 release, since the first steps are to decide on the build environment and get everyone's toolchains synced. You can find the release schedule at <https://discourse.ubuntu.com/t/jammy-jellyfish-release-schedule/23906>, which shows when new software planned for inclusion is released.

Chains of command

Fedora is governed primarily by the Fedora Council, which includes representatives from all over the project, Red Hat-nominated members and a few others. Beneath that are FESCo, the Fedora Engineering Steering Committee, and MindShare. FESCo is responsible for deciding on the technical direction of the project and MindShare is all about community outreach, conducting liaison between teams and encouraging contributors to mix with other teams. Reporting into FESCo and MindShare are many smaller Engineering and Community teams. See the diagram (below left).

Like Fedora, Ubuntu's development releases feed into its big, stable releases (the LTSes) which is what people will run on their servers and will be supported for 10 years (with an Ubuntu Advantage subscription). It's these releases, in conjunction with Canonical's support for enterprise tooling (OpenStack, Kubernetes, Ceph, etc), that fill its coffers, so it's in its interests to make the LTS editions fantastic. Also like Fedora, contributions to each new release don't just come from Canonical employees, but also from other companies and volunteers. Oversight is apportioned between the Ubuntu Technical Board and the Ubuntu Community Council, which are analogous to FESCo and MindShare. A key difference is that Ubuntu has a SABDFL (self-appointed benevolent dictator for life) in the form of Mark Shuttleworth, who has a casting vote on both of these committees.

Ubuntu's Technical Board and Community council both meet fortnightly on IRC. This might seem like an outdated (or, if you've never used IRC before, complicated) way to do meetings. But it's better than Zoom and,



since it's how a great deal of open source projects are co-ordinated, is unlikely to go away soon. Fedora boards also use IRC, and have a great guide for newcomers at <https://fedoramagazine.org/beginners-guide-irc>.

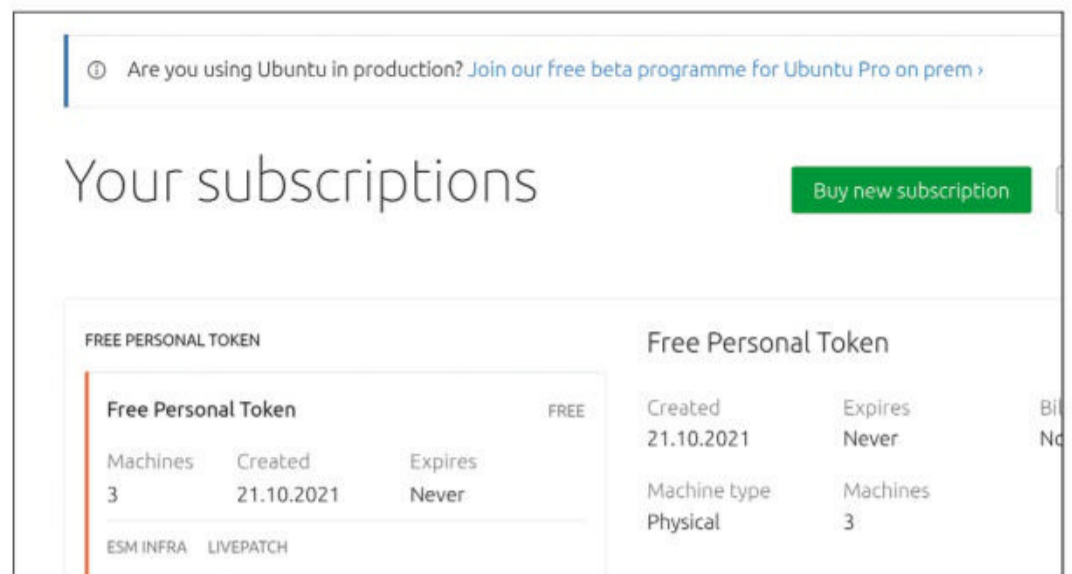
Perhaps an overlooked part of making a distro is baking the files (ISO or USB images) that people will download and install. Fedora and Ubuntu both have their own tools for doing this non-trivial task.

Support levels

Both Canonical and Red Hat make money from providing support to their Enterprise customers. For machines on your own infrastructure, Canonical offers Ubuntu Advantage for Infrastructure (UA-I) and a quick glance at <https://ubuntu.com/pricing> shows that comes in three tiers. Its cheapest Essential support offering is available to those using the LTS releases on servers (\$225 per machine per year), virtual machines (\$75) and desktops. This doesn't include phone support, but you can pay extra (\$525 per annum) for that, and is only available for their LTS offerings. If you're looking for paid support with installation, drivers or anything else consumer oriented then this isn't really for you. It's more aimed at infrastructure and server management (using Canonical's *Landscape* administration tool).

That said, the Essential tier is available for free on up to three machines (or 50 if you're an Official Community Member), and includes Extended Security Maintenance for older releases (namely 14.04 and 16.04) and access to the kernel live patching service. Again, kernel patching is only available for the standard LTS kernel, so no use for Ubuntu 21.10. Fedora is a community supported distribution, so there's no paid support there. Red Hat supports Red Hat Enterprise Linux, which technology tested in Fedora (and now CentOS Stream too) will eventually find its way into once it's stabilised. If you want support (beyond updates) with Kubernetes or OpenStack then you need to move up to the Standard tier. If you're running Ubuntu on a public cloud, i.e. AWS, Azure or GCP, then you can, for a tiny bump on your hourly costs, switch to Ubuntu Pro instead.

For community support, there are official Ask Fedora and an Ask Ubuntu websites (run on Discourse and StackOverflow, respectively). These both receive dozens of questions a day, and both have easily accessible information on how to ask good questions and be a



Ubuntu Advantage is free to sign up for, but for home users the benefits are limited. Livepatching won't obviate update-related reboots when running a GUI...

good human. There are also more traditional forums at The Fedora Lounge (<https://forums.fedoralounge.com>) and <https://ubuntuforums.org>. If you think you've found a bug then familiarise yourself with the bug-reporting process for each distro, so that the developers can fix it. If you're a beginner, or angry, then please resist the temptation to post until you're familiar with this process or have calmed down.

For tracking bugs Fedora uses the popular Bugzilla application where as Ubuntu uses the equally popular Launchpad. Both platforms have much the same workflow: bug reports are triaged, tested and (hopefully) fixed, but they might also be marked as "Won'tFix" (where the issue is judged not to be a problem) or as a duplicate of another bug. The Bugzilla application is also used for feature requests, but again you should familiarise yourself with the etiquette here. Launchpad enables more advanced users to file 'blueprints' for desired features.

It goes without saying that you should perform due diligence and check your bug (or feature request) hasn't already been reported before filing a new one. Fedora maintains a list of common bugs at <https://fedoraproject.org/wiki/Bugs/Common>, which you should scrutinise for the aberration you're planning on reporting. Sometimes the same bug will manifest itself in different ways, so inevitably some seemingly different bugs will end up being marked as duplicates.



» STOP BUGGING ME

Besides manually reporting bugs, you can also send crash information to the relevant teams. Fedora's Automatic Bug Reporting Tool (*ABRT*) will spring into action whenever a Fedora-packaged application crashes, and automatically sends an anonymised crash report to its Abrt Analytics server. This (without human intervention) collates similar reports and if a solution is available the user is given appropriate instructions. If no fix has been identified reports will be looked at by packagers who hopefully will come up with something soon.

ABRT can add stack traces as well as prompt the user for details. Ubuntu's *apport* does much the same, reporting to the Ubuntu Bug Control and Bug Squad teams. Study these reports at <https://bugs.launchpad.net/ubuntu/+source/plasma-workspace/+bug/1945904>. There are a few relics listed there so sort them by number to see what's new in the world of Ubuntu crashes. For more details on reporting Ubuntu bugs, or joining the bug squishing teams, check out <https://help.ubuntu.com/stable/ubuntu-help/report-ubuntu-bug.html.en>.

Sometimes bug reporters will be invited to test packages from each distro's proposed updates repository. These are easy to set up, but in general it's a bad idea to enable them universally, since this might end up upgrading every package, for which there's a proposed fix. This makes it very likely that, even if some bugs get fixed, other packages will break. Fedora's Bodhi system (<http://bodhi.fedoraproject.org>) is used to track bugs with proposed updates, where as for Ubuntu everything is more or less unified on [Launchpad.net](https://launchpad.net).

Harness your hardware for smoother streaming

Use your fancy graphics card to decode videos and free up your CPU for your other calculations and compositions.



Since browsing the web constitutes a great deal of most people's computing time, we thought we'd see if we could find any subtle differences between each distro's out-of-the-box Firefox configuration. We've said before that Firefox can be made smoother by enabling the WebRender backend and activating VA-API for hardware-accelerated video decoding. But getting this to work in the real world takes a bit of trial and error.

Both distros ship Firefox 93, and if you do a fresh install of Ubuntu this uses Mozilla's Snap. Users upgrading from previous Ubuntu versions will get the DEB version from the repos. The first step is getting VA-API working, which on Ubuntu was pretty easy. A simple `sudo apt install vainfo` pulled in all the required video drivers. Then running `vainfo` and not getting an error message showed that iHD (the MediaSDK driver for newer Intel graphics) was

ready for VA-API processing. On Fedora the process was harder. The package containing `vainfo` is called `libva-utils` on Fedora, but just installing this (with `dnf install libva-utils`) was not enough – no drivers were pulled in. We searched the base repositories for `libva` and found `libva-intel-hybrid-driver`, which turned out to be no use – that driver is for older chips and we had a 10th Gen Dell XPS. So a bit of Googling evinced that the required drivers were available in the RPMfusion repositories.

RPMfusion hosts various popular packages that can't be included in Fedora's stock offerings. There are two RPMfusion repositories, `free` and `nonfree` and they're really easy to enable nowadays. So easy you don't even need to type anything: just browse to <https://rpmfusion.org/Configuration> and click the "RPM Fusion free" link for Fedora 35 (or whatever version you're running). Firefox on Fedora is already set up to open RPM files with Package Installer, so follow the prompts (don't worry about the "missing security signature" warning) and the repository will be added. Inside the free repo you'll find the `libva-intel-driver` package, which contains the i965 libva-accelerated driver.

» PLODDING TOWARDS ACCELERATION

Back in 2010, some enthusiastic fellow filed a bug at the Mozilla bugtracker asking for HTML5 video acceleration in Firefox. That seemed like a reasonable ask, seeing as it was already implemented for other platforms (well, Windows at least). The bug report was closed in 2019, but you can still read it at https://bugzilla.mozilla.org/show_bug.cgi?id=563206. Or there's another one, still open, at <https://bugs.launchpad.net/ubuntu/+source/chromium-browser/+bug/1424201>. The first few responses show that this was never going to be an easy thing to implement. And our efforts over these two pages show that even with all the bits now in place (technically they've been there since Firefox 78), getting everything working is far from straightforward.

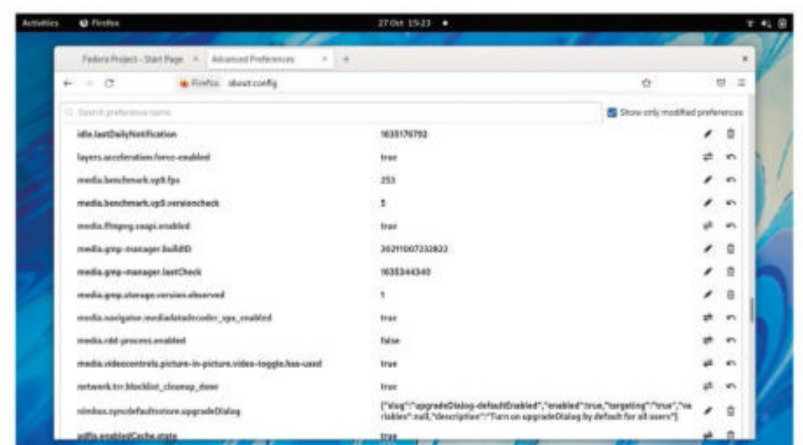
You might think that the situation was better in Chrome/Chromium, and you would be right, but only just. VA-API acceleration for Linux did make it into Chromium via an unsupported patch in 2018, which was later included in some distro packages. The option has been available in official Chrome (and Brave, Vivaldi and Opera) since 2020, too. But Chromium is only available as a Snap on Ubuntu. And guess what? That Snap doesn't yet support VA-API. There's an experimental build you can try with:

```
$ sudo snap install chromium --channel=candidate/vaapi
```

but we couldn't make it work, even after starting with the `--enable-features=VaapiVideoDecoder` option and spending hours messing around in `chrome://flags`. The RPMfusion repos for Fedora host the `chromium-freeworld` package, but we didn't have time to go down another rabbit hole.

Intel Inside, but is it working?

This works for all but the most recent (Icelake and above) Intel chips, but on Gen8 graphics (Broadwell) and above you may want to try the iHD driver. This contains some proprietary bits, and as such can be found in the nonfree RPMfusion repo. Add that in the same way as we added the free one (you can't enable only RPMfusion-nonfree). Then install the iHD driver with a swift `sudo dnf install intel-media-driver`, and a quick look at `vainfo` will show if you're in business. To decode the video you'll also want the `ffmpeg-libs` package from RPMfusion-nonfree.



There are all sorts of settings we can change in Firefox's `about:config` section, but it's generally easier to break things here than fix them.

Actually getting VA-API playback enabled in *Firefox 93* took a bit of headscratching. Fortunately the same options worked on both distros, so we'll summarise them here. Open **about:config** in *Firefox* and set the following options to **true**:

```
gfx.webrender.all
media.ffmpeg.vaapi.enabled
media.navigator.
mediadatadecoder_vpx_
enabled
```

The last one will speed up VPx-encoded WebRTC traffic, which is what's used for video chatting.

There are a number of posts online that suggest other options, but this is a fast-moving area, so some of these will be out of date. Some contained bad advice to begin with – in particular, anyone telling you to disable *Firefox*'s RDD (remote data decoder) process by unsetting **media.rdd-process.enabled** or via the **MOZ_DISABLE_RDD_SANDBOX** environment variable should be ignored. The current situation is that the RDD sandbox blocks VA-API, causing a **seccomp sandbox violation** error if you start *Firefox* from the terminal. But turning it off altogether is a security risk, so don't do that. Instead, selectively disable it for video decoding by setting **media.rdd-ffvpx.enabled** and **media.rdd-vpx.enabled** to false.

You can check if it's working by starting *Firefox* with this doozy:

```
$ MOZ_LOG="PlatformDecoderModule:5,Dmabuf:5"
firefox
```

This generates a huge amount of output to pore over and will probably overrun your terminal's scrollbar in no time. Try pausing the video after a very short time to check you're not missing some early error messages. If you see messages like **VA-API releasing dmabuf surface** among it then this is indicative of success. Different hardware supports different formats, in particular decoding AV1 (denoted AV01 in the Stats for Nerds overlay in the YouTube player) requires very new hardware (newer than our 10th Gen XPS). VP8/9 and h264 (AVC in YouTube) video are more widely supported. For new hardware using the iHD driver it's currently required to set **security.sandbox.content.syscall_whitelist** to 220 and start *Firefox* with

```
$ MOZ_SANDBOX_ALLOW_SYSV=1 firefox
```

in order to let a required syscall out of the sandbox and not upset the iHD driver.

If all that seemed like hard work, rest assured that getting VA-API working with the *Firefox* Snap (the default browser on Ubuntu) proved much harder. In fact, at the time of writing it seems downright impossible because the Gnome platform Snap is missing those driver libraries we mentioned earlier. A fix is on the way, but if it hasn't landed yet you can uninstall the Snap and use the version from the repositories without issue. Just use the options we've given here, some combination of them will surely work.

There's another reason you might not want to use the *Firefox* (or *Chromium*) Snap in Ubuntu, particularly if



you're a fan of Gnome extensions. Cast your eye at <https://bugs.launchpad.net/ubuntu/+source/chromium-browser/+bug/1741074> and you'll see that both browsers refuse to load the native host connector plugin. That's what the website **extensions.gnome.org** uses to manage your Gnome extensions, and without it these can't be managed in the browser. It's not the end of the world – there's a desktop tool called *Gnome Extensions* available in *Ubuntu Software*. Or install it manually with:

```
$ sudo apt install gnome-shell-extension-prefs
```

There, your extensions are yours again, but popular password managers also use native messaging to talk to browser plugins, so if you use those you'll want to avoid Snap-based browsers for now. Rob Gibbon, product manager at Canonical, assured us that there are plans to fix this in time for the LTS release. And this is a healthy reminder that, while annoying, it's good that such bugs get visibility through the interim releases. In general, using a *Firefox* build that comes directly from Mozilla is a good move. One of the main motivations behind Snaps (and Flatpaks) was to enable developers to ship their software independently of distro packagers. It's a sentiment echoed by Rob: "Mozilla wants to deliver *Firefox* directly to the user, which is great for everyone involved. From a QA perspective, we worked with Mozilla to ensure its QA processes met our needs for Ubuntu".

Nvidia and AMD

We tested the **libva** situation pretty thoroughly on Intel graphics, and thanks to the Mesa drivers the situation should be much the same on AMD hardware, or with the Nouveau driver. The proprietary Nvidia driver has its own **Nvdecode** and **Nvencode** API, which is supported on some applications, but not web browsers. There's another API **libvdpau**, again lacking support in popular web browsers, but widely supported by media players and AMD and Nvidia hardware. Interestingly, The Gnome web browser (sometimes known as *Epiphany*) supports all these APIs through *Gstreamer*. As usual, the Arch Wiki is the best place to get the lowdown on all this, in particular the lovely tables at the end of the video acceleration page at https://wiki.archlinux.org/title/Hardware_video_acceleration.

We're not proud of the number of times we watched this video just to get the Video bar in *intel_gpu_top* to read something other than zero.



Flavours, spins, upgrades

Fedora and Ubuntu have all kinds of alternate editions, and both can easily be customised beyond recognition.

Besides the flagship desktop offerings, there are a number of other official editions of Fedora and Ubuntu that you might be interested in. For fun and games (although it actually wasn't fun at all) we tried installing both OSes on an old machine that, with its 2GB RAM and ancient (but still 64-bit) Celeron processor, fell well below the recommended specifications.

Once the installs were complete (which took ages because cheap laptop hard drives of the early 2010s are not fast), both OSes were surprisingly useable. But what turned out to be much more useable was working with the LXQt-powered flavour of each. Gnome had a fairly large, but not surprising given its reputation as the fattest desktop environment, 700MB memory footprint. LXQt had a much more slimline 450MB.

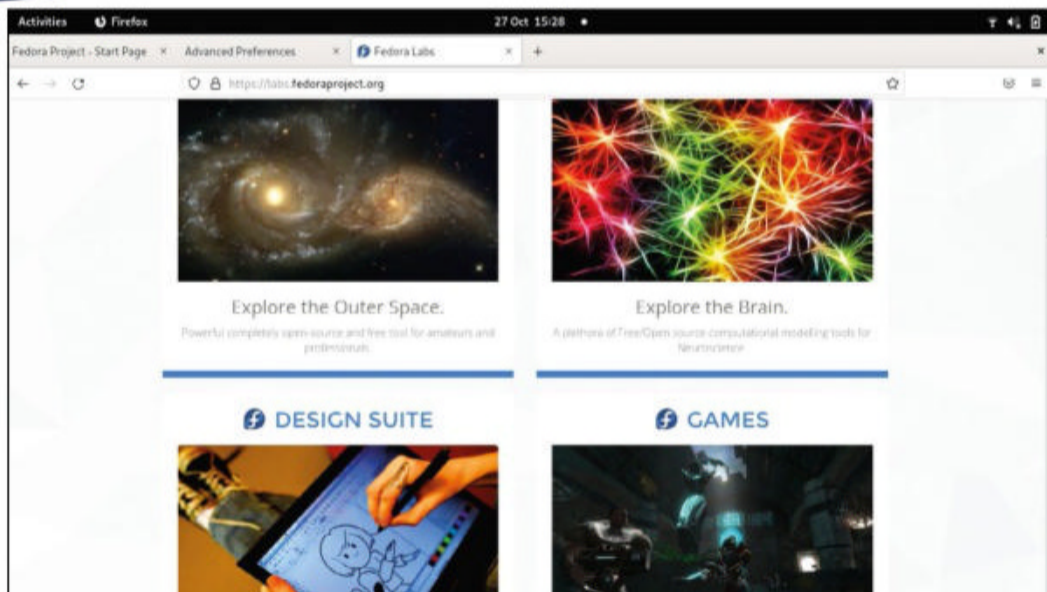
Even if you're not on old hardware, you might not like the Gnome desktop. And if you are, you might like that there's still a Fedora Spin that uses LXDE, the

lightweight desktop built on the venerable GTK2. For a different kind of nostalgia check out the MATE-compiz spin, which brings back Gnome 2 aesthetics with a wobbly windows twist. Kubuntu and the KDE Plasma Fedora spin are among the most popular alternative offerings, but you'll find there's a spin/flavour for any desktop you could care to name. There's also Fedora Kinoite, an atomically updating (like the official Silverblue release) Plasma spin. Similar to the desktop spins are Fedora Labs. These are special editions that cater to a particular area of interest with bespoke software bundles. If you're into computational neuroscience, for example, try the Comp Neuro Lab. It has all of the best open source neural network simulation software (*is that too niche a genre for a future Roundup?* – Ed).

So come up to the Lab...

Most of the Labs are of a scientific theme, but there's also the Design Suite (for art and creativity) and Jam (for music and audio). There's also a Games Lab, which bundles some quite entertaining (though some very old) FOSS titles. It doesn't include any tweaks or drivers, but we'll look at how both distros fair at modern gaming in a moment. Fedora also has Special Interest Groups (SIGs) which aren't necessarily tied to a particular spin or Lab, but whose mission it is to get the software that's the subject of its Special Interest Group packaged and working nicely on Fedora. We were glad to see that i3 (the lightweight tiling window manager that's favoured by developers and fans of keyboard shortcuts) has its own SIG now.

The Ubuntu Flavours are easier to enumerate. There are seven of them, and six of those are desktop flavours. That leaves Ubuntu Studio, which is very much like a union of Fedora's Jam and Visual Design spins. As such



Space, brains, tablets and aliens – there's a Fedora Spin or an Ubuntu Flavour for everyone.

» SWITCH PULSEAUDIO FOR PIPEWIRE IN UBUNTU

Most of the PipeWire subsystem is already present in Ubuntu, but you'll need its *PulseAudio* daemon, and some auxiliary client libraries if you want it to handle your audio. These bits can be retrieved with:

```
$ sudo apt install pipewire-pulse
pipewire-audio-client-libraries
```

Now we reload *Systemd* so that it knows about the new units, and then disable dirty ol' *PulseAudio*. Note there's no `sudo` here that we're still operating on the side. This should mean if it

doesn't work at least only one user's (i.e. yours, sorry) audio will be broken:

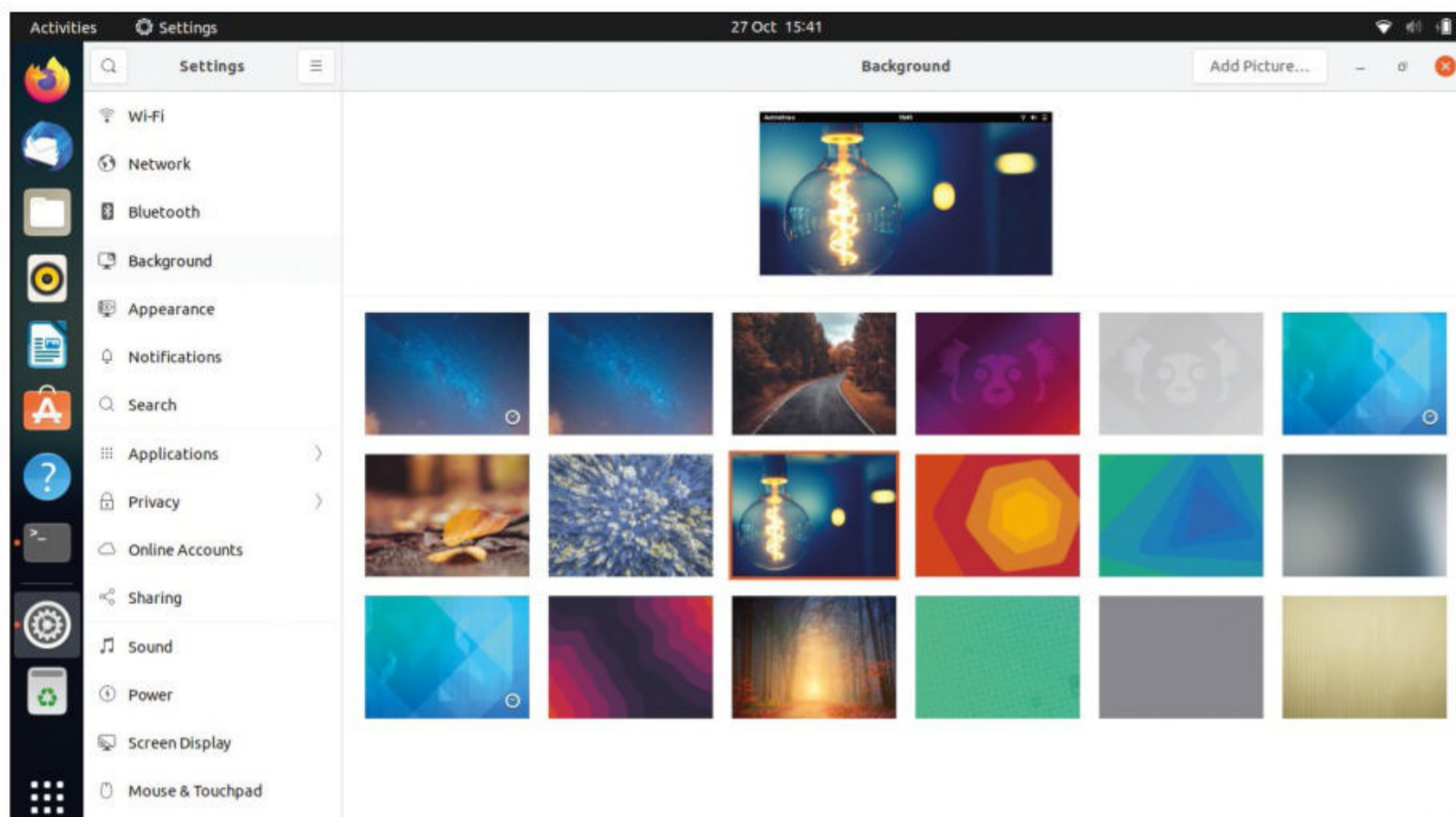
```
$ systemctl --user daemon-reload
$ systemctl --user --now disable
pulseaudio.service pulseaudio.socket
```

And finally we can launch our new audio subsystem:

```
$ systemctl --enable --now pipewire
pipewire-pulse
```

You can check it's working with `pactl info`, the tell-tale line will look like:
Server Name: PulseAudio (on PipeWire 0.3.32)

If you want to try out the latest bleeding-edge version then there's a PPA and instructions available at <https://pipewire-debian.github.io/pipewire-debian>. Because this is new technology, don't be surprised if it doesn't work, particularly on complicated multimedia setups. If it seems like the tooling from the PPA (or indeed any PPA) hasn't had the desired effect, then remember the accurately named `ppa-purge` is available in the repos.



Well it wouldn't be a feature about distros if it didn't have low-resolution grabs of their glorious new backgrounds.

it's a rather large 4.1GB download, which is beginning to push what can physically fit on a DVD (*oh dear—Ed*). Ubuntu Studio used to offer a patched real-time kernel, but since those patches can now be activated in the official kernel sources, and generally cause problems for desktop systems, it's no longer needed. Instead, the low-latency kernel from the Ubuntu repository is used, which should be of a low-enough latency for all but the most fastidious of audiophiles.

On the subject of multimedia, one thing which we haven't mentioned yet is that *Pipewire* is now installed in both distros. *Pipewire* was once described as "*Pulseaudio* for video", which may not have been the best way to advertise it (given many users didn't appreciate *Pulseaudio* when it went mainstream), and certainly doesn't tell the whole story. Instead, one should see *Pipewire* as the key to taming multimedia in an age of Wayland, streaming and screen recording. It's actually the default audio subsystem in Fedora, but unless you went looking you'd never notice anything had changed. That part of it should be a complete drop in replacement for *Pulseaudio*, except in the most edgiest of edge cases. It runs as a *Systemd* user daemon, so to check it's running just enter

```
$ systemctl --user status pipewire-pulse
```

You can get some more information by running `pactl info` (this requires the `pulseaudio-utils` package). If you want to see how it fares in Ubuntu, see the box (*left*). Besides managing *PulseAudio*, *PipeWire* can also handle audio from *JACK*, *ALSA* and *Gstreamer*. It's also designed with sandboxing in mind, so (unlike our web-browser experiments on the previous pages) it works with Snaps and Flatpaks Wayland.

Upgrading your distro of choice

With relatively short lifecycles, you'd expect upgrading to the next release to be nice and easy on both distros. And indeed it is, or at least it should be. We realised that Fedora 35 would be out by the time you were reading this, even though at the time of writing the beta had only just been released. Upgrading to the next release,

beta or no, on Fedora involves just a few steps. First, ensure the system is fully updated with:

```
$ sudo dnf --refresh update
```

The next step is to download the new version, and you'll be reminded that you really ought to have done the previous step before doing this one:

```
$ sudo dnf system-upgrade download --releasever=35
```

Once the new package lists are downloaded you'll have one last chance to bail out of the upgrade. Otherwise it's a case of waiting for a couple of thousand packages to download. In short, a fine time to prepare a cup of tea. When you return, you'll have to confirm importing of the package signing key to the new release and then you're ready to reboot:

```
$ sudo dnf system-upgrade reboot
```

As with regular Fedora updates, the new packages will be installed post-reboot, outside of any GUI. When that's done, which might necessitate another cup of tea, you will be one reboot away from your new version of Fedora. Once you've booted the latest edition, you can easily tidy up any cruft from its predecessor with:

```
$ sudo dnf system-upgrade clean
```

If you've used any of the popular desktop Ubuntu derivatives, you'll be familiar with the friendly alert from the *Software* application telling you that a new version is available. If, for some reason, you don't see this, and you're fully upgraded and rebooted, then there's a new command line incantation to force the upgrade:

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

```
$ update-manager -c
```

And that is largely it for this month's dive into the latest and greatest editions of Ubuntu and Fedora. But over the page you can see what the ever-reliable, straight-talking Mayank Sharma has to say in his head-to-head comparison.



Ubuntu 21.10 vs Fedora 35

Mayank Sharma wonders if there's more that separates the two leading Gnome-based distros than their different packaging formats.

SPECS

Ubuntu 21.10

CPU: 2GHz

Memory: 2GB

HDD: 25GB

Build: x86-64, arm64, armhf, ppc64le, s390x

SPECS

Fedora 35

CPU: 2GHz

Memory: 2GB

HDD: 20GB

Build: amd64 and aarch64

Truth be told, Ubuntu and Fedora, arguably the two leading distros, have very little in common. We apply the fact that both use the Gnome desktop to pit the projects against each other, which only helps accentuate the differences between the two projects and their wares.

In terms of similarities, besides their headline Gnome-based editions, both distros have spins based on various popular desktop environments. Furthermore, both projects support multiple architectures and produce a lot more editions than the ones for the desktop users.

Even though they go about it differently – and we've just spent the last eight pages going over their individual efforts – you can use both Fedora and Ubuntu on all kinds of servers, containers, IoT and public clouds.

In terms of releases, while Ubuntu does put out Long Term Releases every two years, Fedora treats all its releases the same. Ubuntu 21.10 is a standard short-term release that will be supported for nine months, while Fedora 35, like all Fedora releases, will continue to receive updates for about 13 months.

In terms of the usability of the installation and update process, there's little to choose between them. Fedora's Anaconda, and Ubuntu's Ubiquity are two of the best Linux installers. While we can nitpick their different approaches, the fact of the matter is they are comfortable to operate and will get the job done for a vast majority of users.

Known Gnome

Perhaps the most apparent difference between the approach of the two projects is their treatment of their biggest similarity: the Gnome desktop.

Ubuntu 21.10 ships with the Gnome 40 desktop, while Fedora 35 defaults to the latest Gnome 41. While Fedora has already clocked some mileage with the visually uplifted Gnome 40 with its new horizontal workspace

switcher and application launcher, in its previous release this is Ubuntu's first time out with the redesigned Gnome.

That said though, the Ubuntu developers have spent some time and effort tweaking Gnome to mimic several aspects of Ubuntu's discontinued Unity desktop.

For starters, Ubuntu 21.10 takes users straight to the desktop, overriding the upstream Gnome behaviour of bringing up the Activities screen on log in. Another prominent Ubuntu-ness is the placement of the dock on the left-hand side of the screen, thanks to the efforts of Ubuntu developers upstreaming the plumbing required to get the Ubuntu Dock to work with Gnome 40. Then there's the matter of the minimise button, which has been eradicated from upstream Gnome, but continues to be present in windows under Ubuntu.

Talking of the Ubuntu Dock, in the latest release, it now features a persistent trash can icon, instead of a shortcut. Also, very helpfully, the dock now places a noticeable divider between pinned and running applications.

Two of the most prominent Gnome 40 changes that have made it to Ubuntu 21.10 are the overhauled Activities view with its horizontally panning workspaces, and the new multitouch gestures.

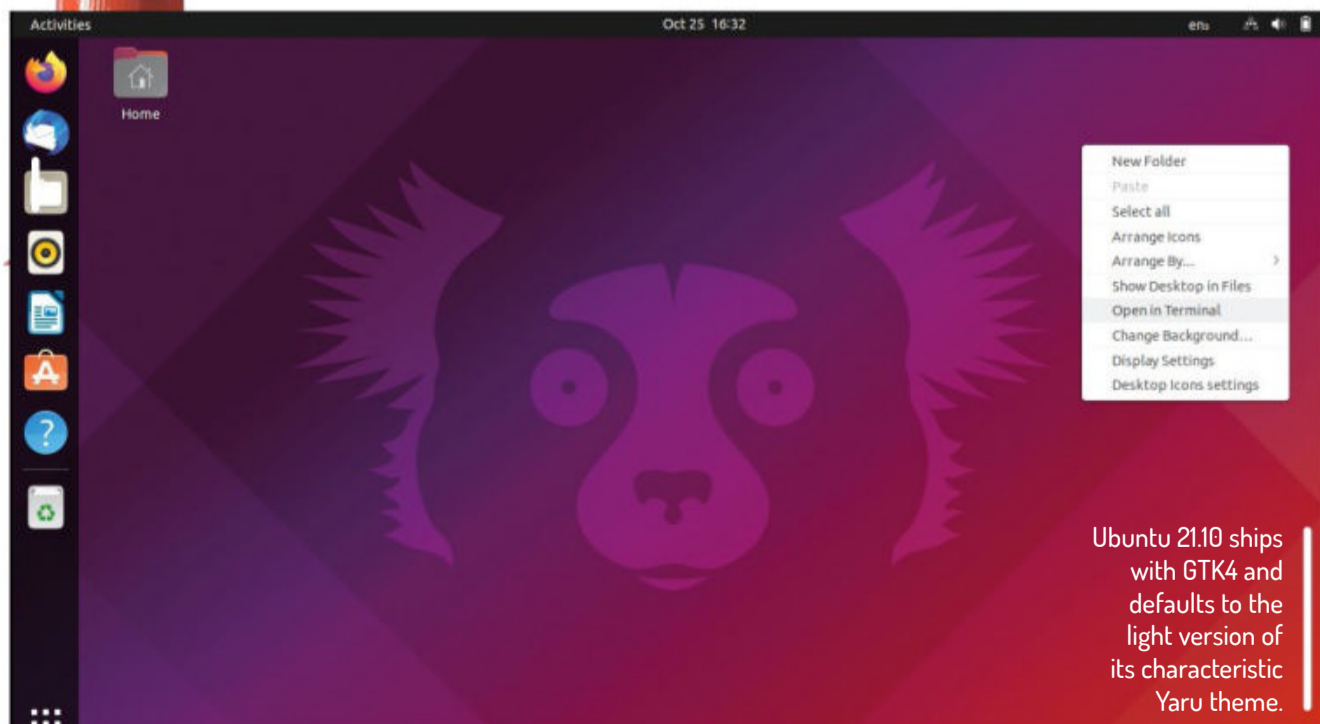
On the other hand, Fedora has long been considered the premier Gnome distro since it virtually always bundles the latest release of the desktop environment. While the launch/dock plays a prominent role in Ubuntu's Gnome, under Fedora 35 Gnome 41 doesn't show one until you bring up the Activities Overview.

Playing with the plumbing

Ubuntu 21.10 ships with the latest version of *PipeWire*, which has caught the fancy of the multimedia world because it can capture and playback for both audio and video with minimal latency. However, the release still defaults to *PulseAudio 15*, which boasts of improved Bluetooth audio codecs.

Meanwhile, Fedora 35, having already given users a taste for *PipeWire*, has now replaced its default session manager with the more feature-rich *WirePlumber* session manager, which allows for more customisations.

One noteworthy settings change under Fedora 35 is the Power settings, which now features a new profile option that enables users to switch between performance, balanced and power-save modes. And while Ubuntu 21.10 doesn't include Gnome 41, it does roll in quite a few programs from its stable, including the *Calendar* tool, *GNOME Disk Utility*, *Eye of Gnome*, *Gnome System Monitor* and more.



In terms of applications, the only notable change in Ubuntu 21.10 is that *Firefox* is now powered by a snap tool, apparently to give *Firefox* developers more control over the update process, though the regular *Firefox* package is still available in the repos.

Fedora 35 gets a redesigned Software store with improved support for Flatpaks, in that enabling third-party repositories will also list selected Flathub applications in the Software store, including *Zoom*, *Microsoft Teams*, *Skype*, *Bitwarden*, *Minecraft* and more.

Breaking cover

Peeking under the hood, it's more of the same. While Ubuntu 21.10 is built around Linux kernel v5.13, Fedora 35 outscores it by wrapping itself around v5.14. The v5.13 became the first kernel to officially include some initial support for Apple's homebrewed Arm-based M1 SoC (system on a chip). It's also the first kernel to introduce support for the Kernel Electric Fence (KFENCE) memory error detector. The feature is enabled by default on Ubuntu 21.10 and will randomise the memory location of the kernel stack at each system-call entry on both the amd64 and arm64 architectures.

One background change that's been due for quite some is enabling the Zstd compression, which will supposedly lead to faster installations. Of course, as you would have probably guessed, Fedora's been using Zstd compressed packages for more than a couple of years.

Oh, and both distributions boast of improved graphics support for users of the proprietary Nvidia graphics drivers, which can now finally run Wayland sessions on top of both distros.

As you can see, the different packaging formats are just one of the many distinguishing elements between the two leading distros. While Fedora 35 gives you an unadulterated Gnome experience, Ubuntu 21.10 strives for continuity by tweaking the desktop according to its sensibilities.

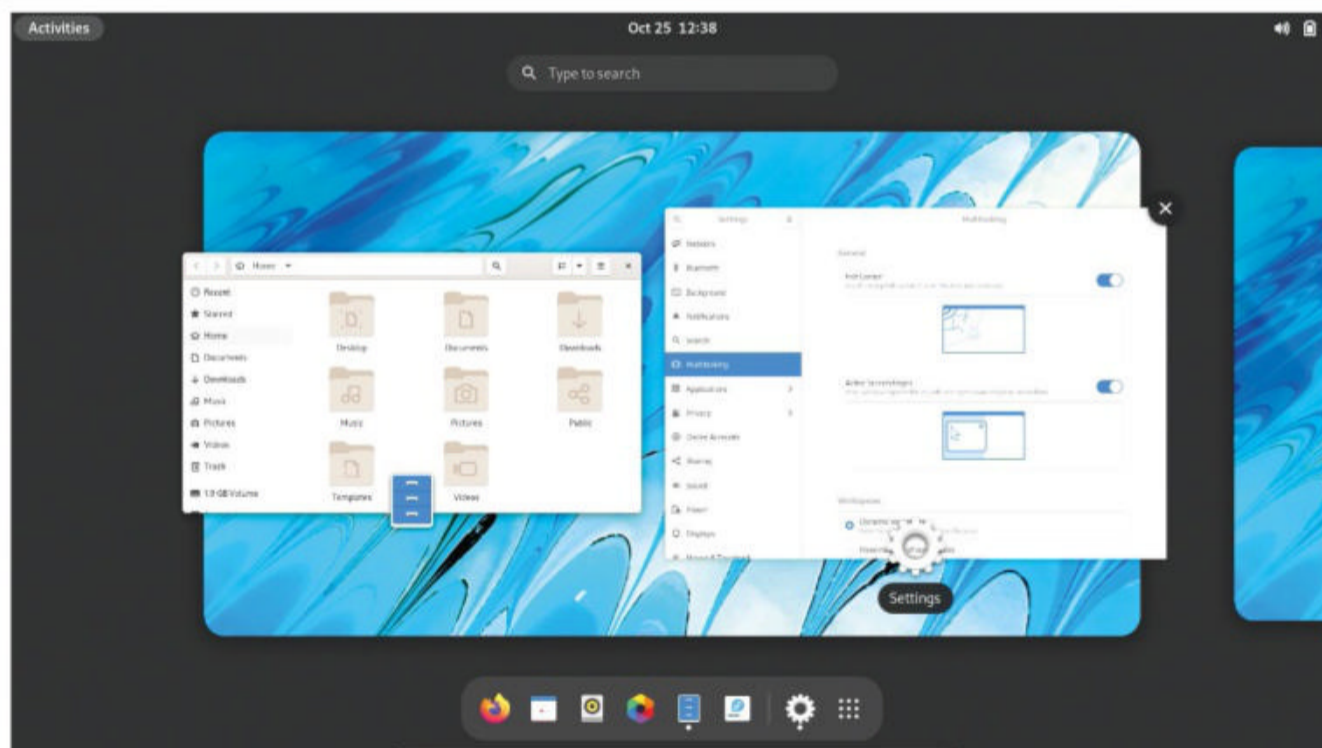
» CALLING ALL CODERS

Canonical hails Ubuntu 21.10 as the most productive environment for cloud-native developers. Arguing that modern development practices rely on containerised images that are consistent and trustworthy, Canonical reveals that with the release it's also published Ubuntu 21.10 images in the Open Container Initiative (OCI) format on the Docker Hub and the Amazon ECR Public Registry.

Other developer-centric highlights of Ubuntu 21.10 include the availability of Apache Cassandra now packaged as a snap, as well as PHP 8 and GCC 11 including full support for static analysis.

Meanwhile, the biggest developer-centric change in Fedora 35 is the distro defaulting to Python 3.10, while retiring the long-unsupported Python 3.5. Elsewhere, the Node.js interpreter has been bumped up to the next 30-month LTS release v16.x, along with PHP 8.

In virtually every aspect Fedora will always be closer to upstream than Ubuntu, and promises a longer support cycle than the standard Ubuntu releases. We can again spend reams breaking down the two distros at a project-level, but from a wares perspective, both distros have equally riveting offerings, making them a compelling upgrade for their respective camps. **LXF**



Fedora 35 gets a new Connections tool, courtesy of Gnome 41, for switching between multiple ongoing remote sessions.

VERDICT: UBUNTU 21.10

DEVELOPER: Canonical
WEB: www.ubuntu.com
LICENCE: Various

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

Ubuntu 21.10 finally makes the redesigned Gnome experience usable, making it a must-have upgrade for non-LTS users.

» **Rating 8/10**

VERDICT: FEDORA 35

DEVELOPER: Red Hat
WEB: www.fedoraproject.com
LICENCE: Various

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

Does enough to maintain its position as the leading Gnome distro that doesn't sacrifice stability in the name of innovation.

» **Rating 8/10**



AUTOMOTIVE GRADE LINUX

Mats Tage Axelsson shows how open source frameworks and Linux can help navigate you safely to your coding destination.



Even though the car industry has been, and still is, quite slow on innovation, cars are now packed with electronics. The systems range from entertainment to keeping track of components and helping you drive safely – or not at all. Ever since Tesla included a tablet instead of instruments, the major manufacturers have been trying to keep up.

The Linux Foundation has joined this race by building the Automotive Grade Linux (AGL) distribution that was established in 2014. Designed to be run in a car (obviously) and available at all times, this means a small, embedded system for

which there are plenty of options. AGL chose Yocto as an underlying build system as it's already aimed at embedded systems and commonly used in IoT and other systems.

Choosing Yocto means all the communication protocols that an in-car system will need are included. Since these are for embedded systems, they already support the majority of buses and interfaces. Included in the distribution are many subsystems including infotainment, instrument clusters and more. As a project, they aim to have the base for all parts needed for a modern car. This also includes telematics and all the

sophisticated stuff an autonomous vehicle needs.

Before you jump to the conclusion that this is a niche project not used in any actual cars, be aware that this is already running in many models from Subaru, Toyota and Mercedes-Benz. The project enjoys wide industry supports as the value to manufacturers is huge, saving them millions in development costs.

Since AGL acts as an overall distribution, many of the parts it utilises come from other projects. As the needs of a modern car are diverse those projects encompass a wide range. We'll cover some of the core ones here.

G ENIVI is a part of AGL and is the project that handles information and entertainment displays. An industry group drives the project, aiming to enable many operating systems for in-vehicle use. They work to allow connections between systems in the vehicle.

One sub-project covers a standard data interface unifying data handling across multiple vehicles. Using standards in this way clears many hurdles for companies and helps them focus on their own special features. The standard also simplifies connection to different cloud providers, which enables competition.

There is a sub-project handling cloud connectivity named CVII. Here you have all the standards that you need to handle your data and connections to the different services that will collect and send data.

Alexa Auto SDK

For the user, you want to have interfaces to the entire range of features in the car. You can choose to use the Alexa Auto SDK to build these functions for clients. Apart from the ordinary features that Alexa supports, you can also use these modules for controlling your heaters, lights and similar parts of a car. Alexa also supports dialling with your connected phone and the text-to-speech module can speak text messages and other communications.

OpenEmbedded

As mentioned earlier, many of the systems in your car will be small and dependent on stability and response. For this, the best standard out there is OpenEmbedded. Using this, you have a straightforward way to support the many platforms that will be available for your in-car systems. OpenEmbedded is widespread in other disciplines like IoT and robotics, making it a very well-tested way of handling your system.

Robotics Operating System

On the subject of robotics, note that included in the set of packages is the Robotics Operating System (ROS – see LXF272). This may seem counterintuitive at first; after all, we are not building Transformers here. When you look through the ROS system, however, it becomes clear that you can get a lot of use from it. With it, you get functions such as cameras, motion sensors and object



Using the map in your car used to mean a separate box, unlike AGL where you have it built-in.

avoidance built-in. You can use all this to make things like Advanced Driver Assistance Systems (ADAS).

Over The Air

Any regular driver of a vehicle will not be upgrading their systems themselves. Over-the-Air updates are an absolute necessity. Here OTA cooperates with AGL and has excellent solutions for this functionality. This functionality is more complex than it seems at first, one key reason being security. It's absolutely vital that updates are correct, and that you protect the download integrity of the update packages. The implications of a failure could mean disaster.

PACKED FULL OF FEATURES

“Functions such as cameras, motion sensors and object avoidance are built-in. You can use all this to make things like Advanced Driver Assistance Systems.”

GERRIT for code review

If you're aiming to develop for AGL you're going to need to learn how to review your code using GERRIT, since

» YOCTO

Yocto is the framework you should use to build the kernel and many other parts of the AGL system. Any embedded system contains the same core components; boot ROM, bootloader, Linux kernel and root filesystem. While this only starts the system, it is absolutely vital for the rest of the system to function properly.

Several projects have ambitions to make a system that covers the complexity of building these four components. AGL chose Yocto for this. Why Yocto, you may ask? One reason is that many industries have honed in on

this project. Other reasons are highly technical and need a few pages of explanation. However, it is powerful while still being flexible, thanks to the recipes used to configure components.

Inside recipes, you have all the data Yocto needs to determine your build. Since the format allows scripts, you can build support for many projects. This flexibility keeps helping developers get the right tools onto their systems. Inside recipes, you point the build to the software you need using the matching URL. This makes it easy to add existing

projects, though getting it to work will take a lot of effort if you are the first one.

```
Firmware Image Package (10)
1 # It is a packaging format used by Yocto to package the
2 # firmware images in a single binary.
3
4 DESCRIPTION = "Yocto - trusted firmware tool for packaging"
5 LICENSE = "BSD-3-Clause"
6
7 SRC_URI = "git://git.yoctoproject.org/yocto-trusted-firmware-a.git"
8 LIC_FILES_CHKSUM = "file://docs/license;md5=18356943"
9
10 # Use Yocto from Yocto 1.8.3
11 SRCREV = "1.8.3"
12
13 DEPENDS += "native"
14
15 inherit native
16
17 do_compile O {
18 # These changes are needed to have the Yocto compilation and exe
```

Recipes is a way to describe a package for the bitbake build system. Note that you point to URLs for projects.

the major projects use this for their work.

GENIVI infotainment

GENIVI is the system that shows you what is going on, mostly entertainment and navigation. GENIVI is the industry standards group that handles in-vehicle infotainment. It started before AGL in 2009 to address the multitude of issues that occur when you try to connect the many systems in modern vehicles.

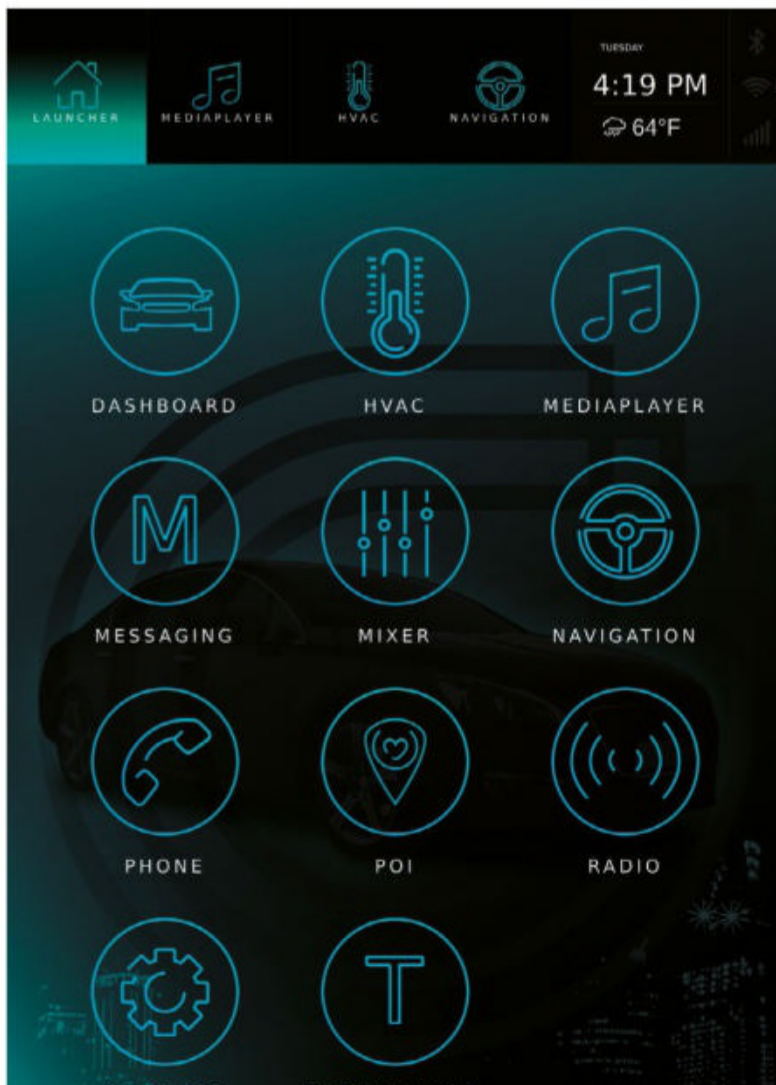
GENIVI works with writing both standard documentation and reference software. Within AGL it is the part of the system to cover all issues of the display, but organised under it are many diverse groups including navigation, speech interface and many more.

Big parts of the system handle Android integration, run by a special interests group. The graphics – 2D and 3D – have standard libraries so that manufacturers only need to tweak for their own products. Moving the system to Wayland has already been undertaken and

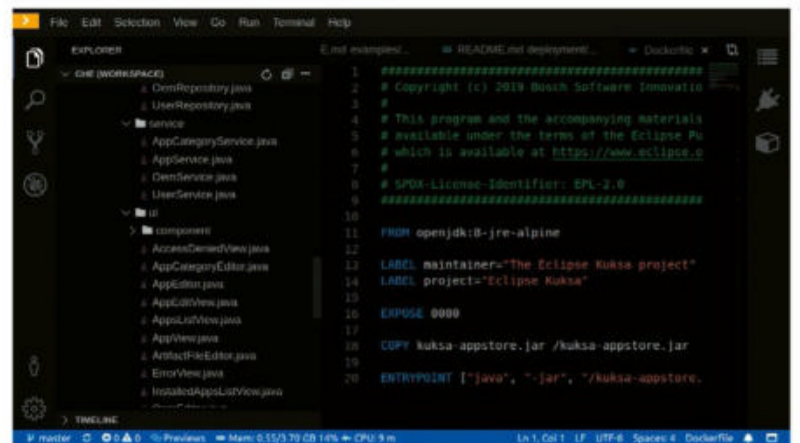
WE NEED A BIGGER CAR

“The directory, after synching, will be around 800MB. Don’t let this fool you. Once you start a compile, it will take hours and create near 100GB of files.”

you can find support for using HTML5 on everything, thus having a completely open standard for designing user interfaces. You don’t need to use the included QT design tools if they don’t match your needs. This is the part that you will see when you run an image of the



Running the AGL in QEMU will show you the default styling and the system in action.



Setting up the Kuksa development system in an online editor can help you get a feel for the system and coding.

system in your emulator of choice.

For obvious reasons, it is not practical to rip out everything you have in your own car and switch it for Linux, but you can still see how it works as a system at home. To get started the source code is available for you to download at <https://bit.ly/LXF283genivi>.

To compile the AGL on your own system, you need 100GB of disk space and a beefy processor. However, if you are mostly curious about what you might see on your own system, you can download a pre-built image. The image is available on the same page and you can download it from the command line.

There are several pre-built images available too, covering the BBE board, h3ulcb-nogfx, qemuarm and the Raspberry Pi 4. Using the images also requires much less disk space (and time) than having the entire source code tree and compiling.

You will see that the entire system depends on the Yocto project, current version 'dunfell'. Using Yocto for this is useful since it already caters to so many needs of the modern car. Connecting to sensors, handling data and supporting advanced AI features is a true strength of the Yocto project.

Getting serious

When you get really serious, you will need an entire development environment to support it all. This requires a build directory of approximately 100GB. The initial download is not very large since there are scripts to fill up what you actually need for your project.

You also need to add *repo*, a code revision tool. Designers, mostly from Google, wrote *repo* in Python to support handling many Git repositories in one project. It does not replace Git. Instead, it helps you handle many interdependent Git repositories. To install on Debian- and Ubuntu-based distributions, use your package manager. For other distributions, you can use a very simple script. In fact, it comes as a single binary which does not need root privileges. As with Git, *repo* will need some control files in the controlled directory. You add those with the `repo init` command.

You’ll notice that the directory, after synchronising, will be around 800MB. Don’t let this fool you. Once you start a compile, it will take hours and create near 100GB of files. This will not surprise developers, but beginners may be, so be aware. You can throw these files away after each compile, but that would be unwise. When you change something and re-compile, the compiler will use most of the files will as-is, saving you vast amounts of time. You can also run an online IDE to do this job. In most cases, though, you will have to get a paid account

with such a vast development tree.

Yes, you can have the entire system on your old Raspberry Pi 4. You even have a ready-built image to run, in case you do not want to set up the development environment. Remember, the big difference here is that you are compiling a kernel for the Raspberry Pi 4. Those parts of the distribution are all in the Yocto source code tree. If you have Yocto experience, this will be a breeze. If not, you have a learning curve to climb.

With that said, you have everything set up from other developers and users, so the tree is there. A warning that you may not notice at first is that they tested the tree with Ubuntu 18.04 in mind. If you have any problems, they may be hard to handle, considering there is no official support for newer versions.

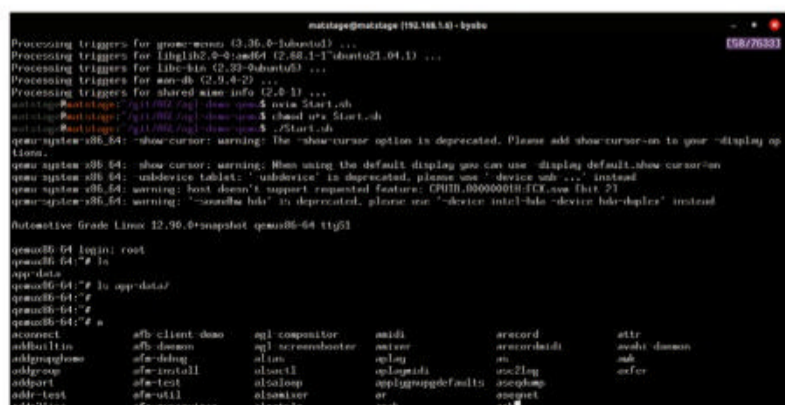
In case you have a never version, it may be better to use a Docker image for your own setup. This makes it clearer and more certain that you'll have the requirements correct. Note that they based the image on Debian 10, so that is what they usually test against. The build script (<https://bit.ly/LXF283docker>) does a brilliant job of creating your environment, but you will save nothing on performance.

Getting serious

The Raspberry Pi may not be the board that car manufacturers want to use, but they have options. For the work of keeping the distribution up to date, project members use a few reference boards. These are versions of the RCar Gen 3 boards. The Renesas corporation built these boards specifically for use in automotive applications. When you get serious about building in-vehicle-infotainment systems, then you can purchase (around £500) these boards. Until then, the Raspberry Pi is the most common and easiest to find.

Images for QEMU complement those two selections, which makes it easy to get started even without buying new hardware, assuming you have a decent current machine. Apart from those officially supported boards, the community also supports a few others. Barring the ARC HS, all boards are Arm boards, supporting both 32-bit and 64-bit platforms. You have the BeagleBone board, the i.MX 6 and 8 and the Snapdragon (410c and 820c).

Inheriting from webOS, AGL uses the Wea Application Manager for developers to create applications. If you are so inclined, use the sample applications from <https://bit.ly/LXF283samples> to get familiar with how to do it. The general idea is to use web technologies – HTML5, JavaScript and so on – to create your desktop graphical environment. Qt is included, but you still get a wide range of possibilities without being totally dependent on



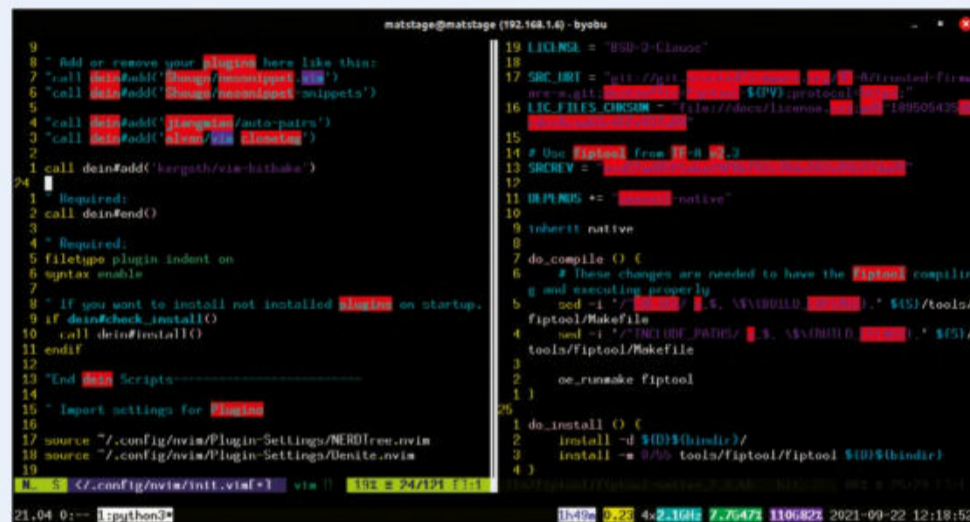
When you start the QEMU emulator, you also get a command line prompt just like on your Linux laptop.

VIM-BITBAKE

When you get really serious about developing for your car, you will need to understand how Yocto works. To configure packages in this embedded software stack involves making recipes for the *bitbake* compiler system. The files have a fairly simple syntax, which makes it easy to sneak in silly mistakes. You can add a plugin called *vim-bitbake* to *vim* to make sure you keep the syntax in check and don't end up chasing such simple faults. It supports simple scripting that can cause headaches and knowing that the syntax is correct makes things much easier.

To install the package is really simple as it exists in your regular package manager – or you can use most plug-in managers directly from the GitHub page (<https://github.com/kergoth/vim-bitbake>). As well as checking your syntax, the plug-in also helps you create new files using templates built-in. Once you get used to a package like this, you will know how useful it is to have the nitty-gritty details checked as you type.

Combine this with the correct plug-ins for your programming language and while you can still make a lot of mistakes, you'll at least know they are serious ones...



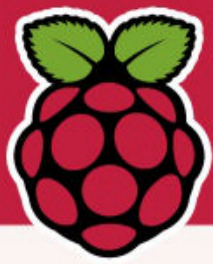
The vim-bitbake plugin is easily installed with your favourite package manager or plug-in manager. Using it makes things much easier.

the underlying platforms.

The examples are very simple; they include an aquarium, *Tetris* and similar applications that are a bit of fun but mostly demonstrate what you can do with the framework. For more in-depth information about webOS, use their website www.webosose.org. Fun fact: webOS runs on many television sets, those from LG in particular, so once you learn, you will also be able to add applications to your TV.

The AGL project has made large strides in the last years and has now a great following in the industry, giving happy enthusiasts a chance to see what is going on in their cars. You are in an enviable position being ready to put some effort in to feed your curiosity about what actually happens in your car systems.

With such well-spread support of the project from both developers and industry, this project will go far and, who knows, actually establish a standard. That may make it possible to allow car owners to repair their own vehicles without losing all consumer rights or their shirts on car insurance. Since you can test all your changes in a small device on your desk, there is nothing stopping you from doing your own testing without leaving the garage. Leave the self-driving to the pros, but the entertainment stuff should be fair game. **LXF**



Michael Horne is the co-organiser of Pi Wars (see piwars.org). Follow the competition on Twitter: [@PiWarsRobotics](https://twitter.com/PiWarsRobotics).

» CHALLENGE ACCEPTED

Pi Wars is a non-destructive, Raspberry Pi-based robotics competition that has both autonomous and remote-controlled challenges. The event is open to absolutely anyone and includes teams of students, families and hobbyists as well as solo roboters, who all build their competing robots either from kits or custom parts.

The main goal of Pi Wars is to provide a way for young and old to get involved in coding, electronics, robotics and the Raspberry Pi. The first Pi Wars competition was held in Cambridge, England in December 2014. Since then the annual event has attracted international interest, with teams entering from all over the globe. Normally an in-person event attracting hundreds of attendees, we moved online in 2020 due to the Coronavirus pandemic. With the success of our "At Home" competition in 2021, we're taking a similar route for 2022 where our theme is farming and agriculture. Teams will build their robots and their own arenas and then video themselves taking on their challenge courses.

You'll be able to view a live stream of the challenge-run entries on 9–10 July, 2022. If you want some robotics action right now, the 2021 videos are online now (<https://piwars.org/2021-vpw/streams>). If you'd like to apply to take part in the 2023 competition (which we hope will be in-person!), simply join our mailing list to receive the official announcements. We also have a Discord chat server for you to talk about robotics to experienced and novice roboters.

Raspberry Pi suffers supply chain issues

The first-ever price increase and the reintroduction of smaller memory models helps fill the gaps.

Eben Upton took to the interblogs to explain a couple of awkward changes to the usually cheap-as-chips Raspberry Pi range. The key points are that the Pi 4 2GB model is temporarily increasing in price from \$35 to \$45, and the reintroduction of the discontinued Pi 4 1GB model at the \$35 price point. This gives buyers the option of hitting a price point, albeit with less memory.

Upton explains this is down to supply chain problems post Covid-19 in the semiconductor industry, which are likely to run through the whole of 2022. Interestingly, these are hitting older 40nm-based products the hardest, which means anything that isn't a Pi 4, Pi 400 or Compute Module 4 will be in short supply.

To offset this shortage, Eben stated, "In allocating our limited stocks of 40nm silicon, we will prioritise Compute Module 3, Compute Module 3+, and Raspberry Pi 3B,

and deprioritise Raspberry Pi 3B+." The thinking here is that tasks intended for the 3B+ can be easily achieved on the Pi 4 1GB model. Meanwhile, Eben is confident that the Pi Foundation has enough 28nm silicon to meet demand and maintain price points on those products.

Read more at <https://bit.ly/lxf283piprice>.



Pricey pandemic Pis put punter's poor pockets into poundless poverty.

Vulkan on Pi Fancy Pi graphics drivers.

Phoronix noticed the Mesa Git V3DV driver for the Broadcom VideoCore (used in the Raspberry Pi 4) was quietly updated and now has Vulkan 1.1 support. This brings better compatibility with DirectX 12, gives it multi-GPU support at the API level and supports ray-tracing, geometry shaders and advanced GPU compute functionality. It'll also better play *Quake 3*, which is nice. Find out more by visiting <https://bit.ly/lxf283vulkan>.



Vulkan support brings a host of advanced features.

CREDIT: Sascha Willems

Pi extreme! Running at 3GHz.

Claude Schwarz has overclocked his Pi Compute Module 4 to an extraordinary 3GHz. Schwarz took a Compute Module 4 sitting in a PiTray mini carrier board, removed the heat spreader, applied a liquid metal thermal paste and replaced the heat spreader. To reach the magic 3GHz Schwarz had to use ice spray. More details at <https://bit.ly/lxf283clocker>.



You could overclock your Pi...

CREDIT: Claude Schwarz

Raspberry Pi Zero 2 W

Meet **Les Pounder's** all-new pocket rocket, complete with updated CPU.

SPECS

CPU: Broadcom BCM2710A1, quad-core 64-bit SoC (Arm Cortex-A53 @1GHz)
RAM: 512MB LPDDR2
GPIO: Standard 40-pin (unsoldered)
Ports: Mini HDMI, Micro USB 2.0 OTG, CSI-2 camera connector
Video: h264 1080p30 encode/decode
Comms: 2.4GHz 802.11b/g/n Wi-Fi with PCB antenna, Bluetooth 4.2, Bluetooth Low Energy
Size: 65x30mm

There have been three models of Raspberry Pi Zero. The original \$5 model first appeared in 2015 and lacked a camera connector, which was rectified in an updated model. But we had to wait until 2017 for Wi-Fi and Bluetooth capabilities to reach the Zero. The Raspberry Pi Zero W was a little more expensive at \$10, but worth it for the wireless connectivity.

Six long years after the initial launch, we're finally seeing a CPU update to the Raspberry Pi Zero range. The Raspberry Pi Zero 2 W is a \$15 quad-core Arm Cortex A53, similar to that of the Raspberry Pi 3. In fact they share the same BCM2710A1 die, but the Zero 2 W is underclocked to 1GHz on all cores to reduce heat output.

The form factor is identical to all of the previous Zero models. In this case the lack of change is for the better because it means we can drop the Pi Zero 2 W into our existing projects and cases. Indeed, we checked the official Raspberry Pi Zero case and it fits like a glove.

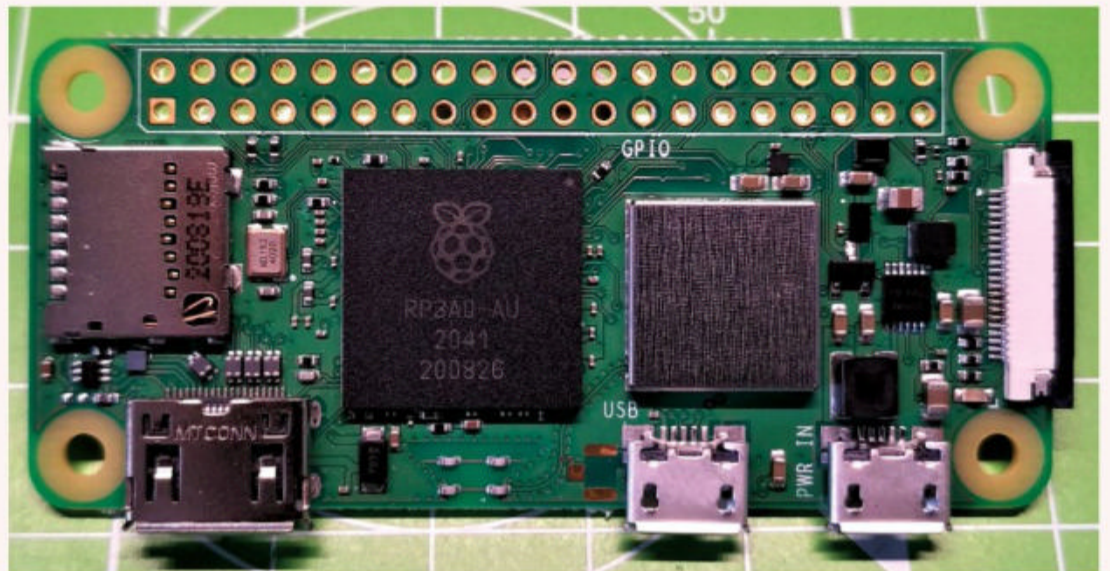
Power consumption

The improved performance of the Pi Zero 2 W does see it draw a little more power, which is something that you'll have to factor into your projects. With the help of *Stressberry* we found that the original Pi Zero W draws around 260mA at idle and 370mA when stressed. Running the same test on the new Pi Zero 2 W produced 280mA at idle and 580mA under stress.

We installed the current version of Raspberry Pi OS 32-bit, and proceeded to power on and test a series of HATs. First on the list was Pimoroni's Explorer HAT Pro, a board that this author is very familiar with; it was used to teach students in over 25 Picademy training sessions. Explorer HAT Pro installed and ran without a hitch, as did the pHAT version.

Streaming video on YouTube proved to be a disappointing experience – in our tests the device failed to play a clip at even 720p resolution. Performance improved once we connected a USB-to-Ethernet dongle, but we still noticed frequent pauses and stutters. Wi-Fi is competent but do remember that this is a PCB antenna, so you'll need to be close to a strong access point.

At idle, the Pi Zero 2 sits at 36.5°C, compared to the older Raspberry Pi Zero W that idles at 32.6°C. When we used *Stressberry* we saw the Pi Zero 2's temperature rise to a high of 63.4°C, which is much higher than the Pi Zero W's 44.4°C. For reference we ran *Stressberry* on a Raspberry Pi 3B and saw an idle temperature of 40.8°C, and a *Stressberry* high of 80.1°C!



■ A key feature is the new custom SoC – the RP3A0 has been designed in the UK by Raspberry Pi.

Not all of our testing was so much fun. We tested a collection of general operating systems for the Raspberry Pi and many didn't work. Ubuntu 20.04 and OSMC failed to boot, despite using the correct image for the Arm v7 CPU. In contrast, the Lite version of Twister OS ran remarkably well. This is an alternative OS based upon Raspberry Pi OS, but with extra applications for gaming and media consumption. Obviously its Pi OS origins gave it a head start over the other OSes.

For now at least, the best bet for the Raspberry Pi Zero 2 W is the official 32-bit OS, because even the beta version of Raspberry Pi OS 64-bit failed to boot. We contacted Raspberry Pi Founder Eben Upton about 64-bit support and the signs are that this will be something that will become available after launch, by the time you read this.

The Raspberry Pi Zero 2 W continues the legacy of the boards that came before it, but we can't help but notice a creeping uptick in price. Originally the Raspberry Pi Zero was a \$5 computer, then with the release of the Zero W the price climbed to \$10. Now for \$15 we get all that came before it, albeit with an improved quad-core CPU. **LXF**

VERDICT

DEVELOPER: Raspberry Pi

WEB: www.raspberrypi.com/products/raspberry-pi-zero-2-w

PRICE: \$15

FEATURES	7/10	EASE OF USE	7/10
PERFORMANCE	6/10	VALUE	10/10

A great upgrade on the original Raspberry Pi Zero spec. It's a drop-in replacement for Pi Zero projects that require a speed boost, but this isn't a Raspberry Pi for your desktop.

» **Rating 8/10**

SCRATCH

Sensor and motor control with the GPIO

Les Pounder shows how we can take Scratch further with a community-developed tool that builds on our Scratch knowledge.



OUR EXPERT

Les Pounder is associate editor at Tom's Hardware.

YOU NEED

- > Pi model 3/4/400
- > The latest Pi OS
- > HCSR04-P Ultrasonic sensor
- > L9110S motor controller
- > DC motor
- > Micro USB breakout PSU for motor
- > 7x F2F wires
- > 2x M2F wires
- > Internet
- > Get the code <https://github.com/lesp/S3GPIO-LXF283/archive/refs/heads/main.zip>

Over the past few issues we've learnt how to create games, control electronics and create shapes by calculating angles, all with Scratch 3. As good as Scratch 3 is there's only so much that we can do with the GPIO, and the GPIO is the best feature of the Raspberry Pi. So how can we take our Scratch GPIO projects further? Well, for that we need *S3GPIO*.

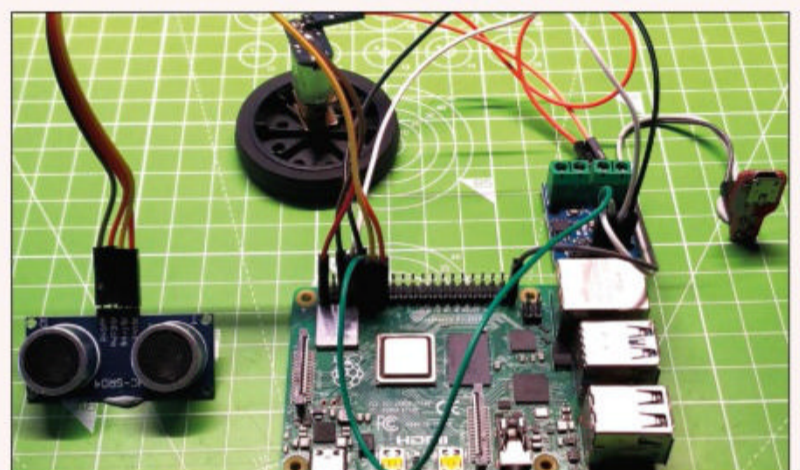
S3GPIO is short for *Scratch 3 GPIO* and it was created by Simon Walters, a Raspberry Pi community member who's campaigned for children to learn coding and electronics with Scratch. Walters first foray into the GPIO for Scratch was *ScratchGPIO*, based upon Scratch 1.4. This version is still active – in fact we can find out more at <http://simplesl.net>.

In this tutorial we'll install *S3GPIO*, learn how to use an ultrasonic sensor as a controller for a DC motor, effectively setting the speed based on our distance from the sensor.

Connecting the electronic components is relatively simple, but the amount of wires may seem confusing. We'll break it down into sections. *S3GPIO* uses physical pin numbering, not the BCM numbering standard set by the Raspberry Pi Foundation. This means that we have two columns of 20 pins. Holding the Pi with the USB ports pointing to the floor, the two columns start with pin 1 at the top left, and pin 2 at the top right. The left column is the "odd" column, with pins following an odd number pattern: 1, 3, 5, 7, 9 etc. The right column is the "even" column, with numbers progressing 2, 4, 6, 8 etc.

Our HCSR04-P ultrasonic sensor has four pins. Using female to female jumper wires, connect GND to any GND pin on the Raspberry Pi, Vcc to 3V (Pin 1). Trigger to pin 11, and echo to pin 13.

Next we need to connect the motor to the L9110S motor controller. Buying a pre-soldered DC motor is the best way to go. Connect the terminals of the motor to MOTORA terminals (it doesn't matter which way) of the L9110S. Screw them into place so they're snugly attached. Now connect GND on the L9110S to any GND pin on the Raspberry Pi. This creates a common ground reference, which is necessary for the project to work. Connect the + of an external 5V micro USB power supply to the VCC of the L9110S motor controller. Connect the – (GND) pin of the power supply to any GND pin on the Raspberry Pi. The final connections are



There are plenty of wires, but by focusing on one section at a time we can easily build up the circuit for this project.

two female to female jumper wires, that connect the A1-A and A1-B terminals of the L9110S to pins 8 and 10 on the Raspberry Pi. This means we can use the GPIO to control the L9110S, causing the motor to whirl into life.

We should now have an externally powered motor, which is connected to the Raspberry Pi. This enables us to control it from *S3GPIO*. We also should have an HCSR04-P ultrasonic sensor connected and ready for us to use as a speed controller. Please consult the diagram for this project to ensure that your connections are correct.

S3GPIO setup

Power up your Raspberry Pi and open a terminal. Enter the following command. Note that this command will download the installer. This is typically a big security risk for production machines, but in this case it's safe.

```
wget https://git.io/vMS6T -O isgh8.sh
```

```
Run the following command to install S3GPIO:
sudo bash isgh8.sh
```

Next, reboot your Raspberry Pi for the changes to take effect.

With the Raspberry Pi rebooted, double left-click the *S3GPIO* icon and select Execute in Terminal to launch three applications. A web server, *ScratchGPIO* and the *Chromium* browser with a special *S3GPIO* extension will all be launched, and it could take a little time until you see *S3GPIO*. Once it's ready click See Inside to open the online editor.

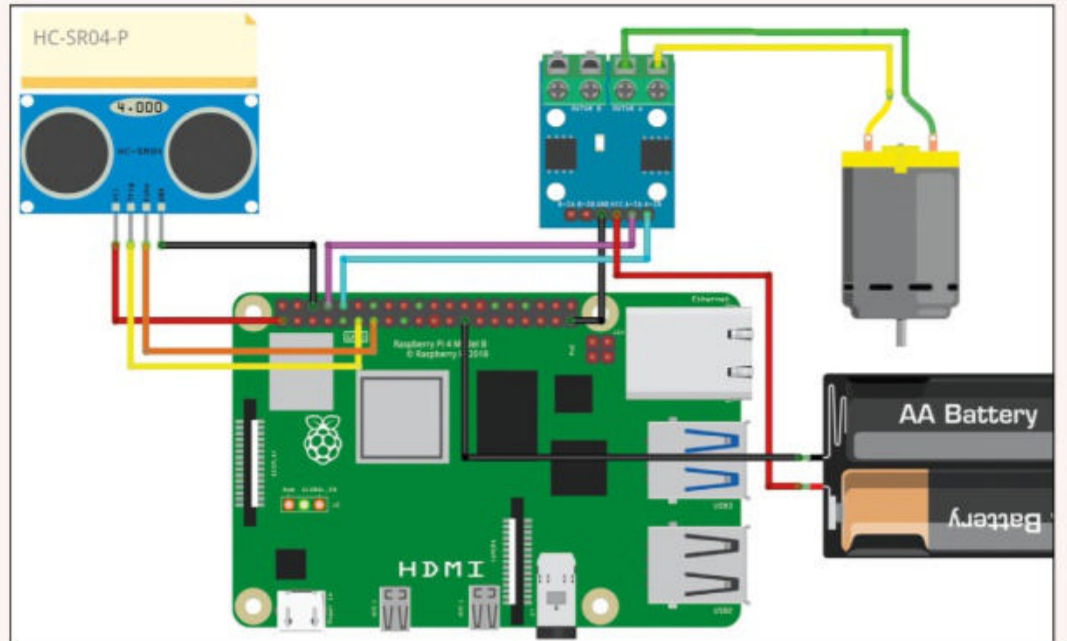
S3GPIO has the same interface as Scratch 3. It also

has a series of *S3GPIO* blocks in “My Blocks” that we shall make use of.

We'll start by testing the HCSR04-P ultrasonic sensor. From the Events section, drag **When Green Flag Clicked** into the coding area. Now from My Blocks drag the **S3GPIO command** and connect it to the previous block. In the white box type `ultra11,13`. These are the pins for the trigger and echo on the sensor. Drag a **Forever loop** from Control and connect it to the previous blocks. From My Blocks drag **S3GPIO read** and place it inside the forever loop. In the white box type `ultra13`. This will read the echo pin of the ultrasonic sensor, and behind the scenes it'll be converted to a measurement in centimetres.

From Looks, drag a **Say Hello!** block and place it under the previous. From Variables, drag **sensor** and place it over the **Hello!**. This will say the ultrasonic sensor distance reading. From My Blocks drag **S3GPIO set _ to _** and place it under the previous block. In the first blank type `Power8`, and in the second place another sensor block from Variables. This will set the motor pin connected to pin 8 to the value returned by the sensor. The last block in this code is a **wait 1 seconds** block, connected to the previous block and edited to wait for 0.1 seconds.

With the code done, click the Green Flag to start. The motor should come to life. Now move your hand nearer to the two “eyes” of the sensor, and the motor will slow. Click the stop button (top right of the stage), but the motor won't stop! We need to make a brake. In a new section of code start with a **When Space Key is Pressed** block, found in Events. Drag this into the coding area. Now from My Blocks drag two **S3GPIO set _ to _** and connect them to the previous. Change the first **S3GPIO “set to”** `Power8` to 0, and the second `Power10` to 0. The last block is found in Control and it's **stop all**. This will halt all the running code. Click the Green Flag and check that the sensor responds to your hand. The motor should go fast or slow depending on where your hand is. When you're ready press Space to stop the motor.



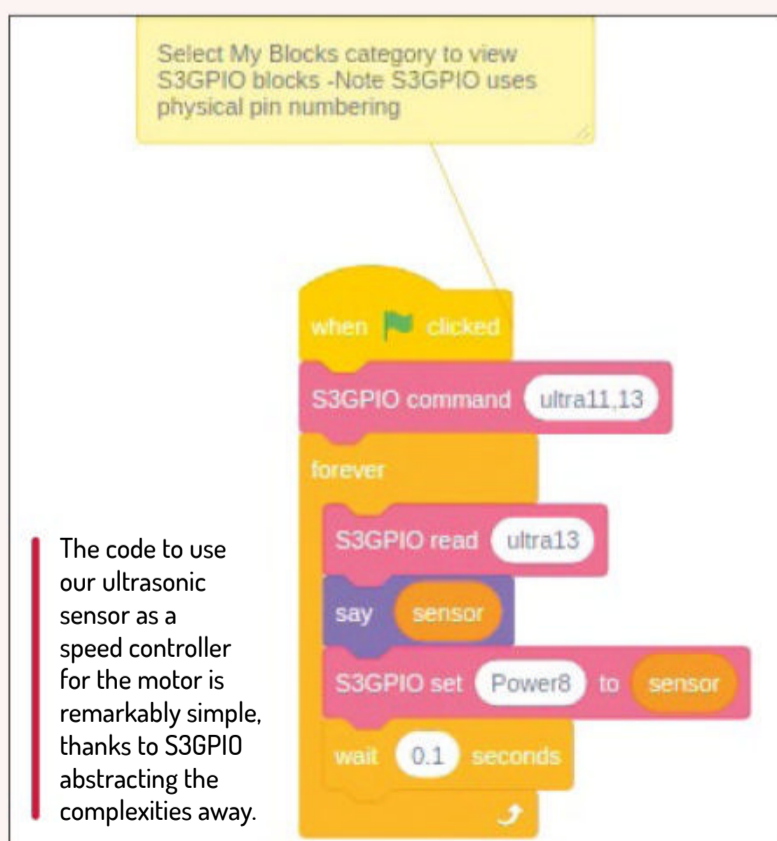
The circuit looks complex, but fear not. The best approach is to break down each section, starting with the HC-SR04-P and then the L9110S motor controller, before moving on to the external motor power.

The final sequence starts with a **When space key pressed from Events**, but change the Space key to an `r`. From Control drag **repeat 10** and place it under the previous block. From My Blocks drag two **S3GPIO** command blocks and place them inside the loop. Change the first to **Pin 8 On**, and the second to **Pin 10 Off**. From Control drag **Wait 1 Seconds** and place it under those blocks. Right-click the top **S3GPIO** block and select **duplicate**. It will duplicate the three blocks that we've just created. Place these three new blocks inside the loop, under the previous. In the next section change **Pin 8 On** to **Off**, and **Pin 10 Off** to **On**. Repeat the duplicate for these three blocks and place the newly duplicated three under them, still inside the loop. Set the **S3GPIO** commands so that both are off. Press `r` and watch the motors move in both directions, then stop.

We've now learnt three ways to control a simple DC motor, something that we can use to control robots and create touch-free user interfaces. **LXF**

QUICK TIP

Need a refresher on how to use **S3GPIO**? **Simon Walter's website** has a link to a **cheat sheet** that covers many of the examples used in this project, plus a few extras: see <http://simplepi.net/basic-guide-to-using-s3gpio>.



» ELECTRONICS AND THE RASPBERRY PI

Electronics projects and the Raspberry Pi go hand in hand thanks to the GPIO. Initially, the Raspberry Pi came with a 26-pin GPIO, but the release of the Model B+ in 2014 saw the pin count increase to 40. These pins can be connected to components, such as the HC-SR04-P and the L9110S motor controller used in this project. The Raspberry Pi GPIO uses 3.3V logic – in other words it only works with components that also work at 3.3V. The HC-SR04-P is a 3.3V version of the original HC-SR04, which ran at 5V (ideal for Arduinos).

The L9110S motor controller is an important component. We can't just connect a motor to the GPIO because it would pull too much current and damage the GPIO pin. Instead we power the L9110S externally from a 5V supply, tying the GND of the power source to the GND of the Raspberry Pi. Then we use two GPIO pins on the Pi to control two input pins on the L9110S. This will trigger the motor spin as we change the polarity of the two input pins, effectively telling the L9110S to send the current in one of two directions. L9110S motor controllers cost only a few pounds, and are well worth investing in for your future projects.

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

ELECTRONICS

Interfacing to single board computers

Mike Bedford shows you how to get more from your Raspberry Pi, Arduino or other SBC by getting to grips with interfacing circuitry.



OUR EXPERT

Mike Bedford
With a degree in computer science, but a long standing interest in electronics, Mike likes to straddle the great divide with a foot in both camps - hardware and software.

For many users, a major benefit of single board computers like the Raspberry Pi and Arduino products is the ease by which they can be interfaced to real-world hardware, and not just computer peripherals. As such, they're commonly used at the heart of all sorts of innovative projects. Many devices can be attached to SBCs via standard interfaces like USB or I2C, and simple components like LEDs and pushbuttons can be connected to GPIO ports with an absolute minimum of additional electronics.

So you might think there's no need to use any external electronics beyond the familiar current-limiting resistor that's necessary for driving a LED, but some devices just can't be interfaced to SBCs without some additional electronics. In other cases, without using some external circuitry you'd risk running out of GPIO pins, or you might have to use a larger SBC than you'd otherwise need to get enough GPIOs.

Here we look at how to use external interfacing circuits to attach devices that you couldn't otherwise connect to your SBC, or to reduce the number of GPIO pins needed. This relates to any type of SBC, not just Raspberry Pis, but whatever you're interfacing to, be sure to check the specification – most importantly the voltage levels and maximum current of its GPIO ports.

We're going to be presenting circuits as schematics, but we'll leave you to figure out how to implement them as real circuits, probably on a breadboard, stripboard or PCB. Several of the components we'll be using are logic ICs in the 74HC family. Different members of the family provide different functionality, but for general info on these chips, take a look at the box on the opposite page.

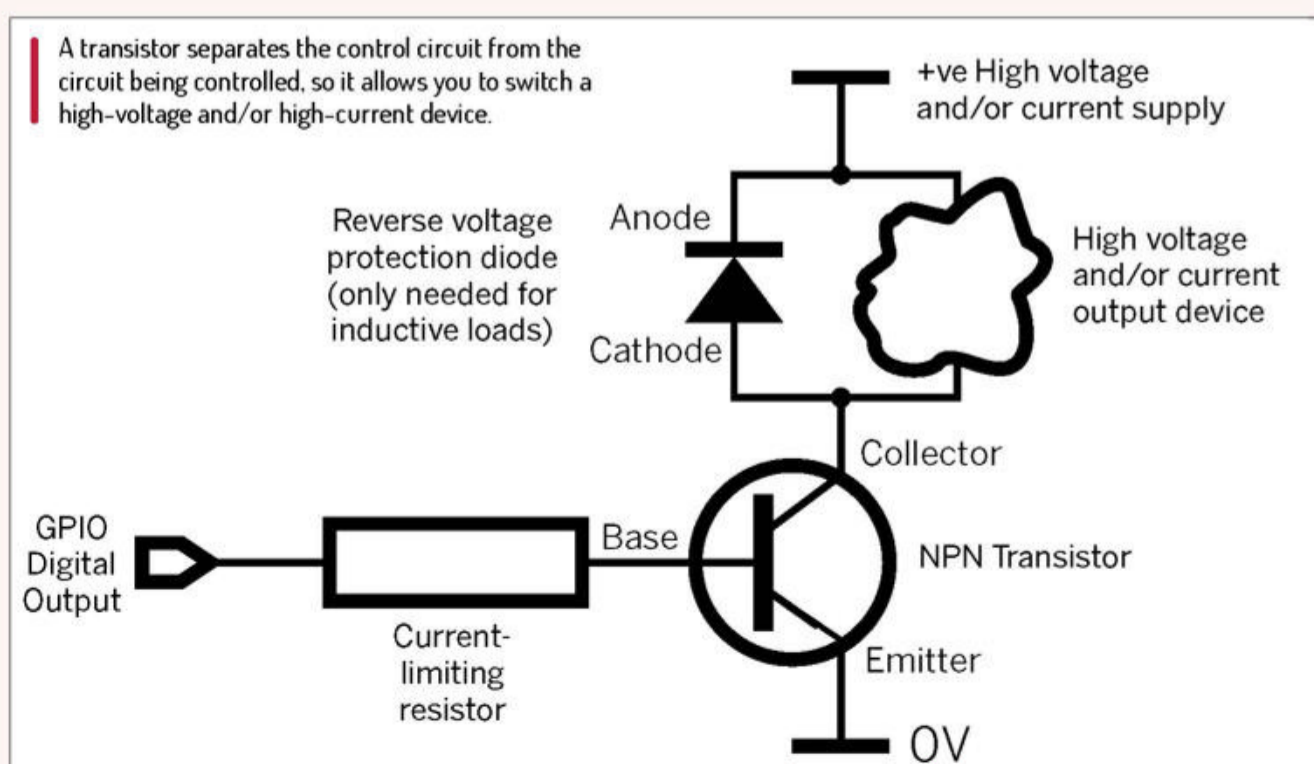
Before looking at specifics, we should raise a general point. Some SBCs operate from a 3.3V supply, others use 5V. Because our instructions are generic, we'll refer to these voltages as V_{cc} . Similarly, irrespective of whether it's referred to as GND or 0V on your SBC, we'll always refer to 0V.

More current, more voltage

Driving an LED from an SBC is a common requirement and it's easy to achieve. LEDs will light with a forward voltage from about 1.6 to 3.1V, depending on the colour, so they can be driven directly from digital output pins that deliver a logic high. They usually work with a current of 10 or 12mA, which falls within the maximum current rating of most SBC's GPIOs (on RPi's, there is a limit of 16mA per pin, or 50mA per GPIO bank). All you need to drive a LED, therefore, is the current-limiting resistor that you're probably familiar with.

However, an indicator LED is one of the few things you might want to drive from a GPIO output pin that can be driven directly. Pretty much everything else either uses a higher voltage than a GPIO pin can supply, or draws too much current. To drive something that's more demanding – say a high-powered LED or a small DC motor – you need to use a bit of interfacing electronics.

Often that circuit will contain just a transistor, a resistor and perhaps a diode, and although you could use either a field effect transistor or a bipolar transistor, we're going to consider the latter. A transistor is a device that allows one electrical circuit to be driven by another. Because transistors have gain, that



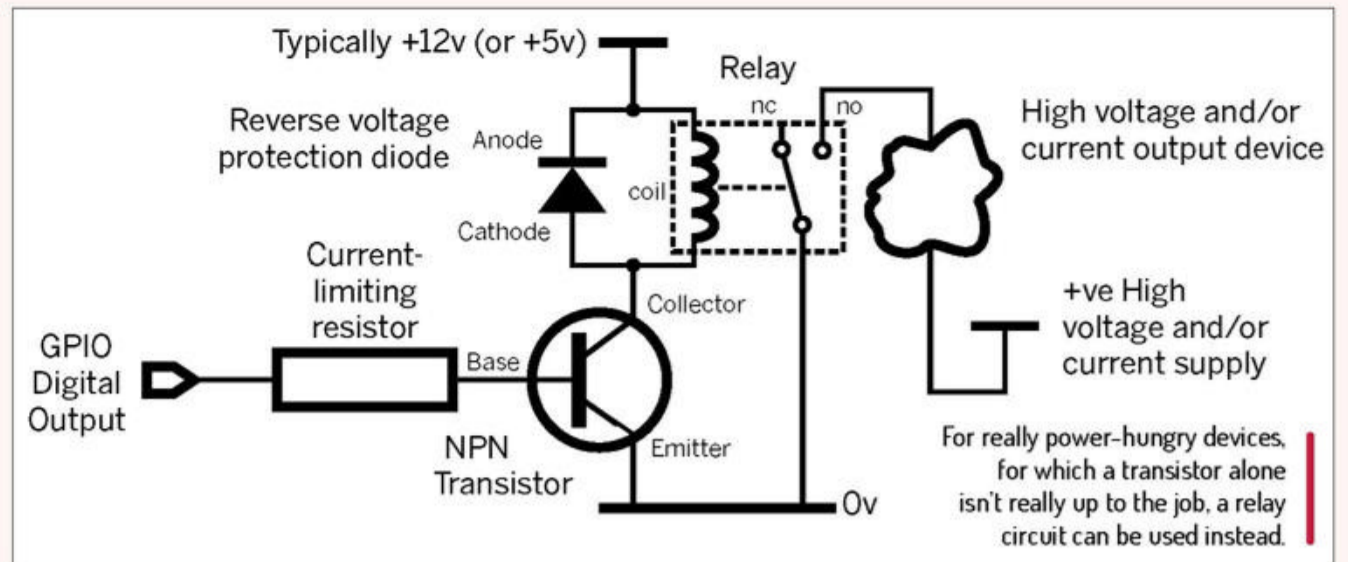
is they can amplify, the current in the device being driven can be greater than the current used to turn it on and off, and because they can usually withstand greater voltages than the SBC's Vcc, the attached device can use a higher voltage than the driving signal.

The image at the bottom left of page 50 shows the necessary circuit. The bipolar transistor is of the type known as NPN, and you can probably use this in most cases, in preference to the alternative PNP type. We've labelled the three

terminals – base, emitter and collector – although these can be identified from the symbol and aren't normally shown on schematics. The driving signal, from a GPIO pin, is applied to the base, and the resistor limits the current. You should calculate the value of the resistor using Ohm's law, and assume that the base is at 0V potential.

To work out just how low you can go, divide the current drawn by the device you're driving by the gain of the transistor, although you'll have chosen the particular transistor bearing this in mind. For example, if the device you're driving draws 200mA and the transistor has a gain of 100, you can use a 2mA drive current, although it's best to use a somewhat higher current so you're not right on the limit. In many cases an NP2222A transistor would be a good choice, but you should look at the datasheet to see if it meets your needs.

The diode isn't needed for a purely resistive load, but with loads that are inductive, such as motors and relays,



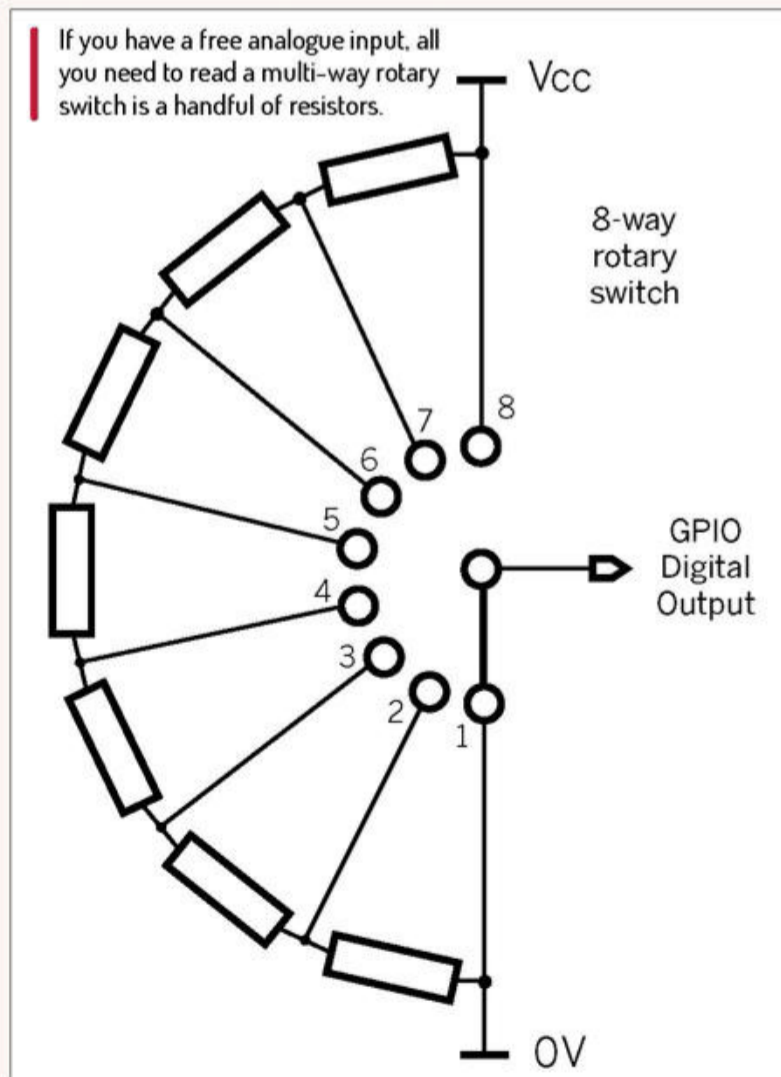
a reverse voltage can be generated when the device turns off. This could destroy the transistor, so a diode, with its anode and cathode connected as shown, is used to prevent this – a 1N4148 will often be suitable. The final thing you need to know is that a positive voltage on the base turns the transistor on and hence also the device being driven, so the code will be exactly the same as if you were driving a device such as an LED without a transistor.

Some loads can't easily be driven by a single transistor because they're even more demanding. If you can't find a transistor with a high enough gain, it's possible to use two transistors in a so-called Darlington pair configuration which has a very much higher gain than the individual transistors. We'll leave you to read up on this, but note that you can make your own Darlington pair from a couple of transistors, or buy it as a package.

Another situation applies to very-high voltage and/or current devices. A more practical solution than a large and expensive transistor, and probably also a heat sink, would be to use a relay. This is an electromagnetic device where the current in its coil – which will be driven by a transistor – produces a magnetic field that attracts and closes a normally open (no) contact, thereby physically completing a high-current or high-voltage circuit – see schematic at the top of the page. There

QUICK TIP

It might be an aside from our main theme, but having now been introduced to 74HC series chips, by using a breadboard plus a few LEDs and pushbuttons, you could use a selection of these chips to get to grips with logic circuitry via a hands-on approach.



» 74HC SERIES CHIPS

74HC chips are an update of the venerable 74-series logic chips that we've chosen because they'll work with either a 3.3V supply and logic levels or a 5V supply and logic levels.

The full name of these parts will be something like SN74HC00N. The first two letters designate the manufacturer, so you can ignore those; the '74' identifies it as a 74-series device; the following letters designate the sub-family of which we're considering only 'HC'; the following figures identify the member of the family, ie. its function (for example 00 is a quad 2-input NAND gate); and the final letter(s) indicate various options, most notably the package type.

Because surface-mounted components are difficult to use without specialist equipment, you'll probably want to use devices in through-hole DIP packages which are designated by a final 'N'. To use 74HC devices, you'll need to scrutinise their datasheets, which usually show a truth table, which show the outputs that result from various combinations of the inputs, and which are easy to understand.

» **INTERFACE WITH OUR SYSTEMS** Subscribe now at <http://bit.ly/LinuxFormat>

QUICK TIP

Although 74HC series chips will operate with either 3.3V or 5V supplies and logic levels, they don't allow you to interface a 3.3V SBC to a 5V SBC, something that would require level-shifting of logic signals. We don't have space to look at this topic, but do bear it in mind if you need to achieve that.

might be more than one (nc) contacts, and usually the same number of normally closed (nc) contacts. 12V is a common coil voltage, although some relays use 5V.

Lots of inputs

Next we're going to look at how lots of inputs can, in some cases, be interfaced using fewer GPIO pins than the number of inputs. A classic example is a multi-position switch, which is a rotary switch that can be thought of as several ordinary switches, of which only one will be 'on' at any time, depending on the switch's rotary position. At one time, these were commonly used to allow one of multiple functions to be selected, although today it's more common to repeatedly push a button to select functions.

However, if you don't want to add a display or lots of status LEDs to your project, or if you want to give your design a retro look, a multi-position switch remains a possibility. The snag, though, is that ordinarily these switches could use up lots of GPIO pins. An 8-position switch, for example, is like eight ordinary switches so,

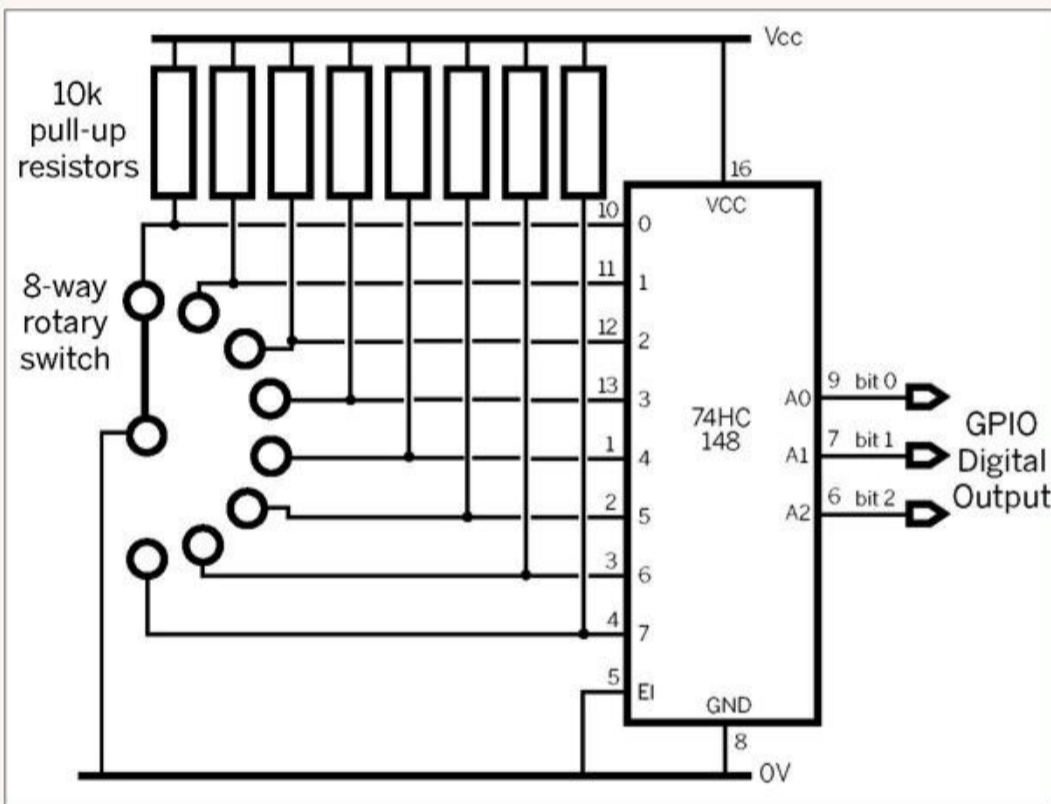
it will require 8 GPIO digital input pins. There are several ways of reducing the number of GPIO pins you need, and if your SBC has analogue inputs (so that excludes RPis) and you have one spare, a simple circuit will enable you to use nothing more than that single GPIO pin. As you can see in the schematic at the bottom left of page 51, the solution is to connect Vcc to the highest-numbered switch position, connect 0V to position 1, and connect identical value resistors between each adjacent pairs of switch positions.

If the switch is at its highest- numbered position, then the switch's common position, which attaches to the analogue input, is at Vcc. If it's at position 1, then it's at 0V. For intermediate positions, the circuit acts as a voltage divider, with the common switch position being at a voltage of Vcc multiplied by the switch position divided by 8.

You can use any value for the resistors, but the smaller the value the greater the current. Note that if you use too high a value, noise could be a problem. For mains-powered equipment, the safest thing is to go for a value of 1k or 2k. For battery-powered gear, you might like to experiment with larger values to reduce power consumption. The voltages will rarely be exactly as calculated for a particular switch position because resistor values are never exact, so the code should allow for voltage ranges for each position.

If you don't have a spare analogue input, an alternative is to use an encoder chip which outputs a binary value depending on which of its inputs is at a given logic level. For an 8-way switch, you'd use a 74HC148 chip, which is an 8-to-3 encoder, so it reduces the number of digital inputs required from 8 to 3. It has active low inputs, so the switch's common connection must be connected to 0V, and unlike most SBC GPIOs, it doesn't have built-in pull-up resistors, so you'll have to add these.

The complete circuit is shown to the left. Perhaps the only additional words of explanation that we need to give is that EI is an enable input, that needs to be connected to 0V in this application, and EO and GS are outputs that we don't use, so they can be left unconnected. Oh, and regarding the necessary coding, note that the 74HC148 outputs an inverted binary value – take a look at the datasheet – but you can easily take account of that in the code.

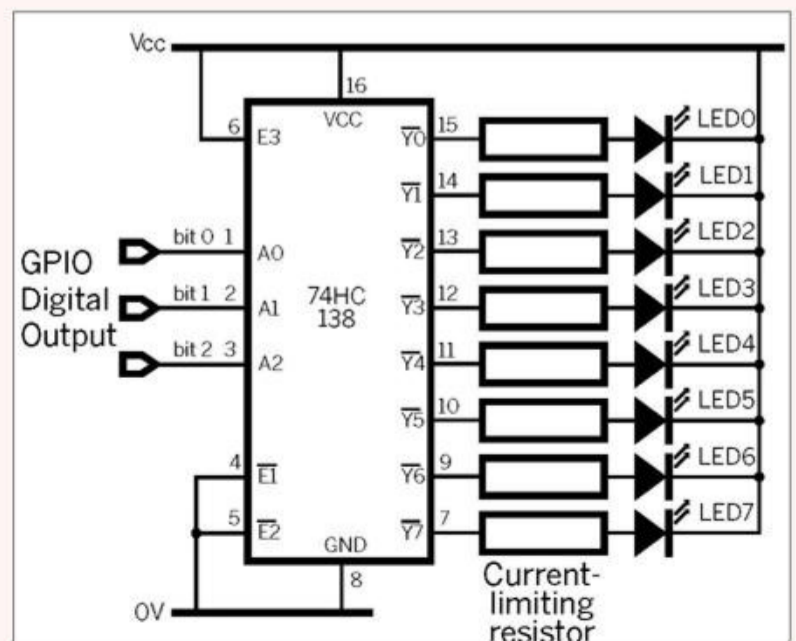


An alternative way of reading a rotary switch without using lots of digital inputs involves using a 74HC148 8-to-3 encoder chip.

» MAINS-POWERED EQUIPMENT

It's possible, using relays or certain types of semiconductor devices that we haven't covered here, to control mains-powered equipment. However, we feel compelled to suggest that, unless you consider yourself an experienced electronics engineer, you don't try building your own interfacing circuit. The potential risks might be obvious but, if not, they include destroying your Raspberry Pi or Arduino, giving yourself a fatal electric shock, blowing up components potentially firing shrapnel into your eyes, or even causing fires.

Don't abandon all thoughts of turning electrical appliances on or off using your SBC, but we recommend only using ready-built modules. If you prefer to play ultra-safe, the most secure option is to use those interfaces which are in fact mains sockets, just like wall sockets, that you simply plug your devices into. Take a look, for example, at the Energenie range of remote control sockets for which a RPi controller, called the Pi-mote, is available. This is a RF device, so it operates from up to 30m away. (<https://bit.ly/lxf283pimote>)



If you want to drive eight LEDs, but only one will be lit at once, a 74HC138 3-to-8 decoder provides a simple solution.

Lots of outputs

We're now turning our attention from lots of inputs to multiple outputs, to see how a bit of logic circuitry can significantly reduce the number of GPIOs required. Mostly we're going to be looking at interfacing LEDs, but the same solutions can be used with higher voltage and/or current output devices, although in that case you'd need to combine the circuits provided here with the sorts of transistor (or transistor and relay) circuits that we presented earlier.

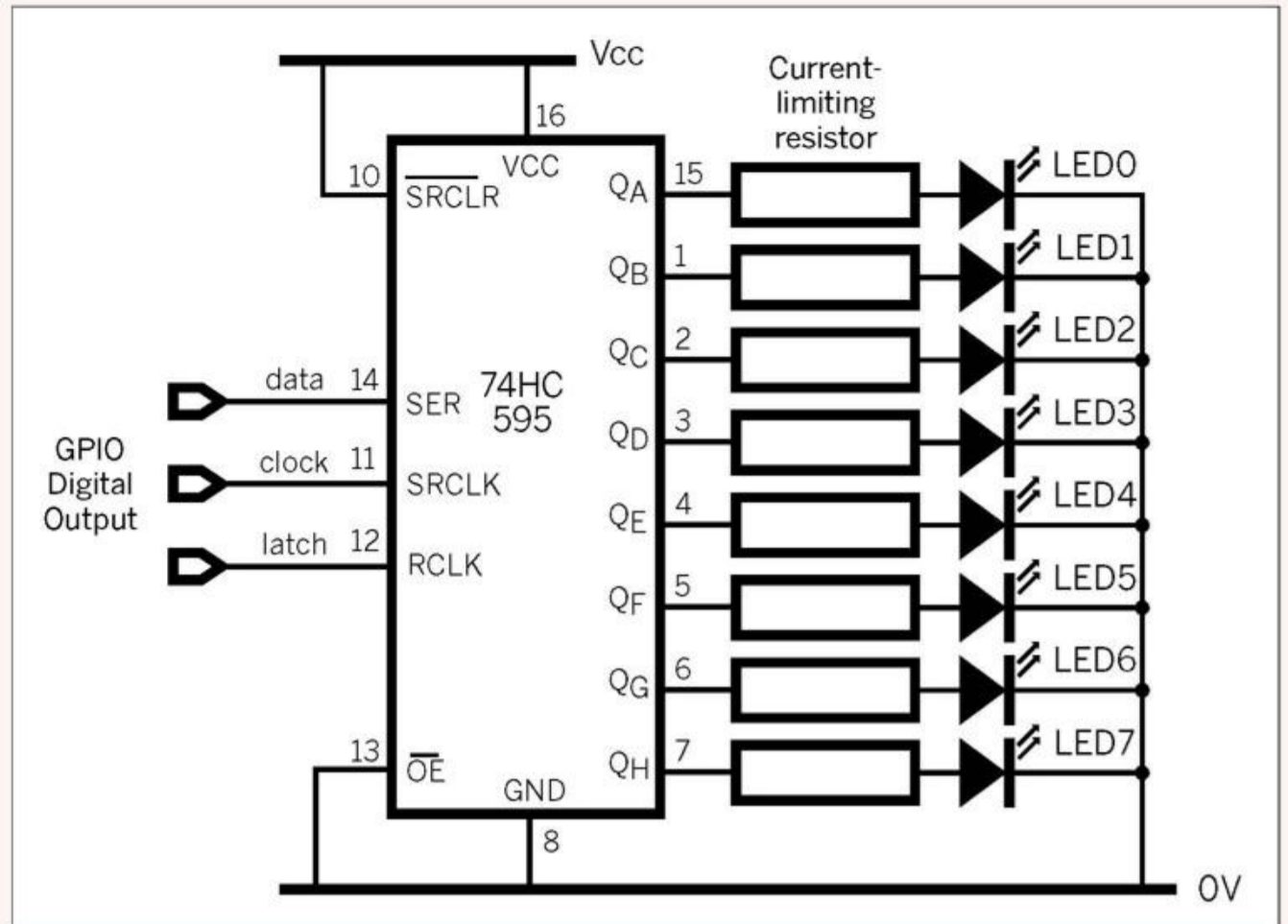
First we'll think about the situation we addressed in the previous section but, rather than lots of inputs, we're concerned with several outputs for which we only ever want one to be active at once. For example, these outputs could be LEDs that are used to indicate a mode that is selected by pressing a single pushbutton repeatedly.

The simplest solution is to do exactly the opposite to what we did with the 74HC148. To take the example of eight LEDs so we're making a direct comparison with the 74HC148, we need a 3-to-8 decoder, instead of an 8-to-3 encoder. The 74HC series chip that performs that function is the 74HC138, and also available are the 74HC139 dual 2-to-4 decoder, and the 74HC154 4-to-16 decoder. Note that the 74HC138 has negative logic outputs, which means that the output corresponding to the binary input are logic lows, so the LEDs driven by those outputs will need to be connected to Vcc. See schematic at the bottom of page 52. Also, there are three enable inputs of which, for our application, E1 and E2 should be connected to 0V and E3 to Vcc.

Although the decoder is probably the most obvious way of interfacing lots of LEDs when only one will be on at once, it's not the only method of driving lots of LEDs while reducing the number of GPIO pins required. The other method requires some slightly more involved code to drive it, but it can even be used in applications where more than one of the LEDs might need to be illuminated at the same time. This alternative method could be used to drive a bar graph display, as used in audio applications to indicate the loudness of a signal.

The solution is to use a device called a shift register which can be thought of as a serial-to-parallel converter. The 74HC596 is an 8-bit shift register and we provide a typical circuit configuration in the image above. You'll notice that there are three inputs that connect to GPIO digital outputs and we've labelled them with meaningful names, specifically data, clock and latch.

The chip contains an 8-bit register or, in other words, an 8-bit memory location; here's how it works. Each time the clock signal goes from low to high, every bit in the register shifts one position to the left, so the value in bit 6 moves to bit 7 (the value in bit 7 being discarded), the value in bit 5 moves to bit 6, and so on, with the final value in bit 0 moving to bit 1. At the same time, whatever value is present on the data input pin is written to the register's bit 0. Any 8-bit value can be written to the



Driving it uses a bit more code than an encoder chip, but using shift registers allows you to control any number of digital outputs from just three digital outputs.

register, therefore, by toggling the clock input eight times while setting appropriate values on the data input for each clock pulse.

This just leaves the latch input. You might expect that the values in the register would be connected directly to the 74HC595's output pins. However, if that was the case, the LEDs would flicker while the register was being written to. So the 74HC595 has an extra input which causes the contents of the 8-bit register to be latched – you should pulse that once you've clocked in all the data – thereby causing the values in the register to be made available on the output pins in place of whatever was there previously.

There's also an active-low output enable pin which, for our application, needs connecting to 0V, and a clear input, which is also active low, so we have connected it to Vcc. Finally, we should point out that there's a bit 7 output so, if you want a register with more than 8 bits, you can daisy chain a pair or more of 74HC595s.

Other interfacing options

To cover this in more details note that Arduinos and some other SBCs have analogue inputs and outputs, but Raspberry Pis don't. So if you want to attach analogue sensors or a potentiometer, or if you want to drive an analogue output device, you'll need to use some external circuitry. You can buy analogue input and/or output HATs, although if you're building something compact, your own external circuitry might be a better approach.

You'll need an ADC (analogue-to-digital converter) to provide an analogue input facility, or a DAC (digital-to-analogue converter) for an analogue output although, in that case, smoothing a PWM output is another option. It's never possible to consider every eventuality, and there are certainly other instances where some external electronics could be used to good effect. **LXF**

QUICK TIP

If you're migrating your circuit from a breadboard to something more permanent like a PCB or a stripboard, you might want to use sockets for your 74HC series chips. This way, if you manage to damage a chip, it's much easier to replace than if you'd soldered it directly to the board.

SERIAL PERIPHERAL INTERFACE

Driving SPI displays

Let's go on an adventure into the SPI protocol, LCD displays and model trains with **Sean Conway** toot-tooting away!



OUR EXPERT

Sean Conway uses Raspberry Pi projects to fulfil his desire to explore electronics while having fun.

SPI is a synchronous serial communication interface specification that was developed by Motorola in the mid-1980s to provide full-duplex (transmit/receive data in both directions) synchronous serial communication between controller and peripheral devices. Synchronous in protocols means that the communication is controlled by a controller (the Raspberry Pi) that talks to the peripheral – in this case an LCD device.

The SPI protocol defines no special bits to manage data. Limiting this overhead allows for high-speed data streaming. To provide some flexibility in communication, the serial clock has two options – Clock Polarity (CPOL) and Clock Phase (CPHA) – which, when configured, establish one of four modes.

In mode 0, data input/output occurs at the rising edge of the clock signal, where the clock polarity is low. For mode 1, data input/output occurs at the falling edge of the clock signal, where the clock polarity is low. To engage mode 2, data input/output occurs at the falling edge of the clock signal, where the clock polarity is high. In mode 3, data input/output occurs at the rising edge of the clock signal, where the clock polarity is high. CPOL and CPHA values are specified as '0' or '1' to form the four unique modes.

Some display boards have no chip select (CS) pin. We must set the SPI mode to 3 for these boards to work. Other boards work with SPI mode set to 0, which is the default for the Pimoroni **ST7789** Python driver we will deploy later in the tutorial.

The SPI Controller generates the serial clock (SCK) and sends it to the SPI Peripherals for synchronising the transfer of data. The SPI Controller always starts SPI communication, because it configures and generates the serial clock signal. A single bit of data is transferred for each clock cycle. The speed at which the data is moved between devices is determined by the frequency of the serial clock. Later on in the tutorial, when the devices are operational, we'll show how changing the serial clock frequency can affect the transfer of data.

The SPI protocol supports one SPI controller and multiple SPI peripherals. The SPI controller selects which SPI peripheral it wishes to communicate with by setting the CS line to a low voltage. When not communicating, the CS line is kept high. The Raspberry Pi supports two CS lines to allow for communication with two SPI peripherals.

The SPI controller activates the SPI peripheral by switching the CS state to 'low'. Data is then sent one bit at a time along the SDO line. The SPI peripheral whose CS is activated reads the bits as they're received on its

SDI line. The most significant bit (MSB) of data is transmitted first. If the controller requires a response then the SPI peripheral returns data one bit at a time, starting with the least significant (LSB) bit of data, using its SDO line.

The other SPI peripherals that aren't activated by their CS line must disregard the serial clock and don't drive their SDO line to send data.

The data command (DC) pin is required by some devices to distinguish between commands or data for the controller. Since SPI only pushes out anonymous bit streams, the user code has to "tell" the controller if this will be a command (say, D/C LOW) or a data (D/C HIGH, for example) bit stream. The BL pin on the display isn't SPI related, but controls the display backlight.

Install the software

Now that you have a Raspberry Pi and display assembled, let's install some software. Open the command-line interface (CLI) display on the Pi and issue the following commands:

```
sudo apt update -y
sudo apt upgrade -y
sudo apt install python-pip python3-pip libopenjp2-7
libtiff5 python3-numpy python3-spidev python3-rpi.
gpio
python3 -m pip install --upgrade ST7789
python3 -m pip install --upgrade Pillow
python -m pip install --upgrade PIL
python -m pip install --upgrade ST7789
enable SPI through raspi-config
reboot
```

The first commands are general housekeeping to those using Pis. The next commands install some prerequisite software needed to proceed. The first *pip* command installs the Python display driver and the next *pip* install the Python imaging software. Support for SPI on the Pi needs to be enabled. This can be accomplished using the native Raspberry Pi menu drive configuration tool *raspi-config*. From the menu items select Interface Option>SPI, and enable SPI with a Yes.

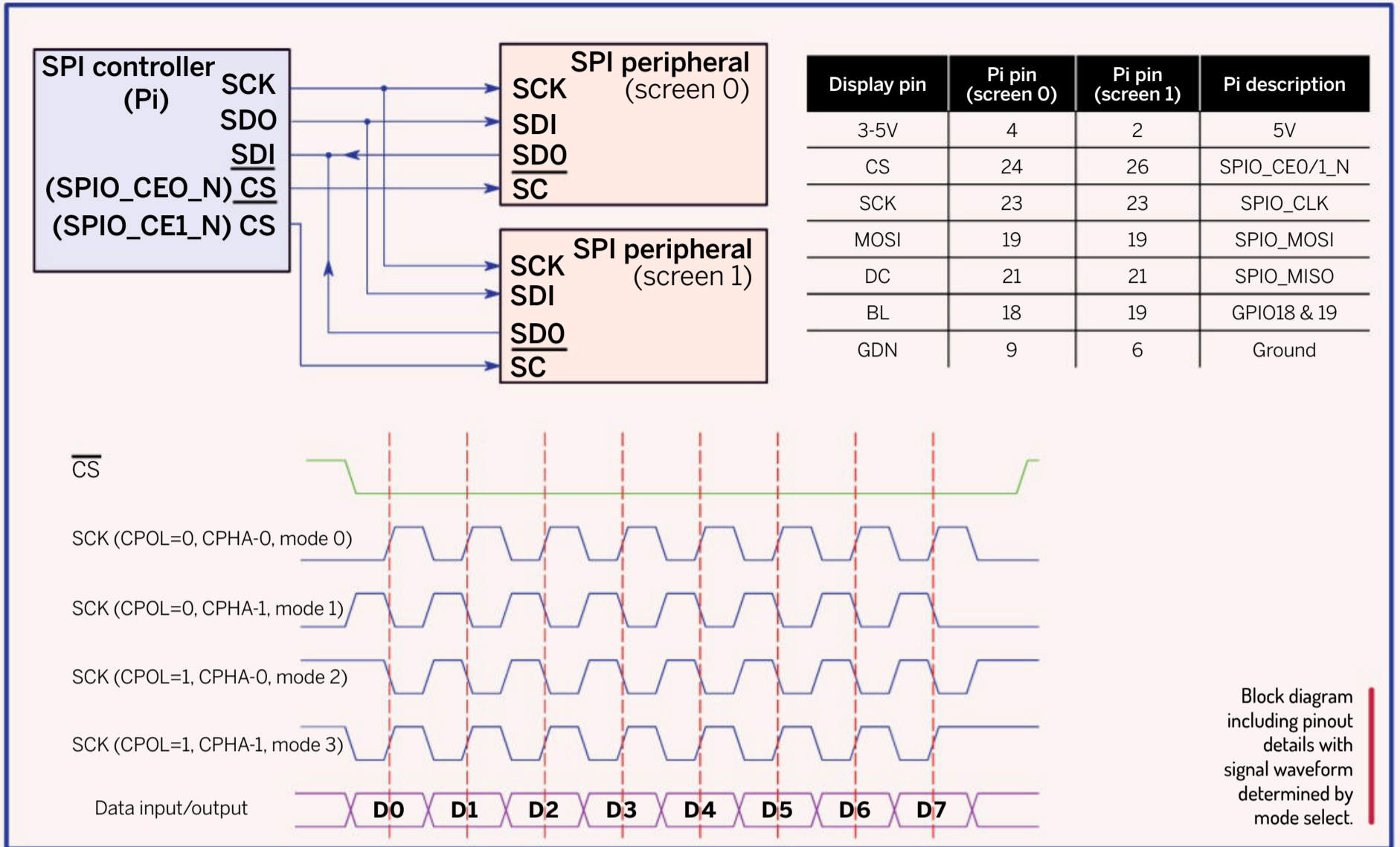
Now that the heavy lifting has been done, let's grab a Python script from the Pimoroni *st7789-python* GitHub repository to test the work. From the link <https://github.com/pimoroni/st7789-python/tree/master/examples> grab the python *image.py* script and the *cat.jpg* image file. Move the python script and image file to the Pi and from the Pi CLI issue the following command.

```
python3 image.py cat.jpg
```

A 1.3-inch feline should appear on the display. Open the *image.py* python script in your favourite text editor

YOU NEED

- > **Raspberry Pi Model 3B+**
- > **SD card 8GB or more**
- > **Pimoroni 1.3-inch SPI colour LCD (240x240) BR (Digi-Key PIM476)**
- > **Raspberry Pi OS Lite (4 Mar 21+)**
- > **Python ST7789 library to control an ST7789 TFT LCD display**
- > **Pillow Python Imaging Library**



and change the script line with the changes shown here.

```
spi_speed_hz=80 * 1000 * 1000, #BEFORE
spi_speed_hz=1 * 1000 * 1000, #AFTER
```

Save the file and run the script. The rate at which the data is written to the screen has been reduced to a level where you can see the image contents being updated on the screen. The block diagram in the SPI section showed two SPI peripherals connected to the same SPI controller. Using the connection legend install screen 0.

To test that the Pi can talk to the displays, run the **image.py** script using Python to confirm screen 1 is working. Open the **image.py** python script in your favourite text editor and change the script lines with the changes shown below:

```
cs=ST7789.BG_SPI_CS_BACK,
backlight=18,
```

Save the file and run the script. The screen 0 should now have the cat image displayed.

Let's explore some capabilities of the Python Imaging Library PILLOW by modifying the **image.py** script. First, we need to add a call to include an additional module from the Python Image Library. Open the **image.py** python script in your favourite text editor and change the script lines with the changes shown below.

```
from PIL import Image, ImageFilter
Locate the text in the image.py file
image = image.resize((WIDTH, HEIGHT))
```

Next, below the line, add the following three image commands, each on their own line:

```
#image = image.filter(ImageFilter.BLUR) #blur image
#image = image.convert('L') #convert image to black and white
```

```
#image = image.rotate(45) #rotate image 45 degrees
```

To test, remove the octothrope (#) in front of each command line, one at a time, and then execute the script. The PILLOW documentation has a handbook containing a tutorial section to explore more features of this outstanding software. **LXF**

» LCD ROAD SIGN MAINTENANCE

The author used the knowledge from this tutorial to design and build an HO-scale model railroad animation simulation. A road sign was constructed using the LCD screen that's controlled by a Raspberry Pi Zero W. The road sign displays historical commercial adverts and the occasional grandchild picture. But it looks like the sign requires some maintenance. Thankfully a repair technician is on the scene, and has climbed on to the platform after arriving in the panel truck.



HO-scale model railroad road sign simulation, complete with dispatched repair technician.

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



LICENCE TO BILL



Jonni Bidwell has been reading licence minutiae and fears big business threatens the future of copyleft and software freedom.

When Richard Stallman wrote the first version of the GPL, even if no one but people contributing to the GNU Project were using it, it represented a major shake-up of traditional licensing practices. Besides defining the oft-quoted 'four freedoms' (numbered 0 to 3, of course), it turned traditional copyright – that is, a list of things you cannot do with a particular work – completely on its head.

Code under GPL was free (as in speech), in the sense that anyone could see and modify it. The only real proviso was that the resulting code had to be subject to those same GPL provisions.

Free begets free. That said, the GPL in no way restricts one's ability to sell software and, optionally, distribute the paid version under a separate licence.

When Oracle was on the brink of acquiring Sun Microsystems in 2009, there was concern that Sun's MySQL database – one of the most successful GPL efforts in history, and one it had acquired the year previous – would effectively become a proprietary product, since any fork of the free work would have to be GPL-licensed. It could not be commercially licensed by anyone other than Oracle, since it – being a company that had a history of making money from databases – inherited the licence. As it happened, this change saw

MySQL founder Monty Widenius release the free MariaDB fork just before the acquisition was approved. Thanks to its compatibility with MySQL, MariaDB is today doing rather well.

But free software's uneasy relationship with industry continues to evolve. In 2018 MongoDB introduced the Server Side Public License (SSPL) and switched its database of the same name's licence to it from AGPLv3. The stated goal here was to stop unscrupulous cloud vendors using MongoDB in their own cloud applications, but pundits were sceptical. To worsen matters the SSPL was later adopted by Elastic Inc. for its immensely popular *ElasticSearch* and *Kibana* products.



Here's a funny thing. We review a bunch of distros every month. And each review has an about box, and each about box has a 'licence' category. Yet, with very few exceptions, it's impossible to say which licence a whole distro falls under. Superfluous page furniture aside, this raises some important questions.

The Linux Kernel has been a GPLv2-licensed affair since very soon after its release. Indeed, it is said that that shift from Linus's original non-commercial licence to a copyleft one directly contributed to Linux's early success. That kernel appears in every Linux distro alongside all kinds of other software. Most Linux distros incorporate GPL-licensed tooling from the GNU project, whence comes the coverline-unfriendly GNU/Linux conjunction. And if that were all they included then licensing would be straightforward. Such a distro would very much be ripe GPL fare.

But distros can contain all sorts of otherwise-licensed software. The most obvious example is the X.org display server, which still powers lots of people's GUIs and for historical reasons is covered by a mishmash of MIT, X Consortium and BSD-style licences. Some distros, like Alpine Linux (and Iglunix iglunix.xyz, almost), don't include any GNU software. Then there's the matter of proprietary firmware and microcode, which lots of modern hardware, from Wi-Fi devices to graphics cards, require in order to light up.

The firmware files themselves are blobs in the `/lib/firmware` directory and run directly on the hardware, typically on a microprocessor of exotic architecture (such as MIPS) for wireless cards. And even if you could convince manufacturers to give you the source files, they wouldn't do you much good because the tools for compiling them are proprietary. Such files are kept in the separate `linux-firmware` package, and typically get



Percona's Head of Open Source Strategy Matt Yonkovit wants databases to stay open. Here here!

loaded during the early boot process. They can also be embedded into a kernel image, which is fine for your own projects, but redistributing such a kernel would be a GPL violation.

On the other hand, there's plenty of 'good' firmware shipped with the kernel, such as the .FUC microcode

WHY FIRMWARE IS FIDDLY

“Even if you could convince manufacturers to give you the source files, the tools for compiling them are proprietary”

files that ship with the clean room reverse-engineered Nouveau driver. Generating these files used to require running the proprietary Nvidia driver and then taking an MMIO dump of GPU registers. So users in search of accelerated graphics would need to do this themselves

» THE LAY OF THE LICENCE LAND

Since licensing is a confusing issue for developers new to open source, GitHub created a helpful website at <https://choosealicense.com>. Besides recommending MIT for those wanting to keep things simple, and GPLv3 for those that want something closer to Free Software ideals, the website has helpful advice on which licences are preferred by what communities. There's also a summary of some of the more widely used licences, such as the Apache License 2.0 (which allows limitations on trademark use) and the “medium-copyleft” Mozilla Public licence.

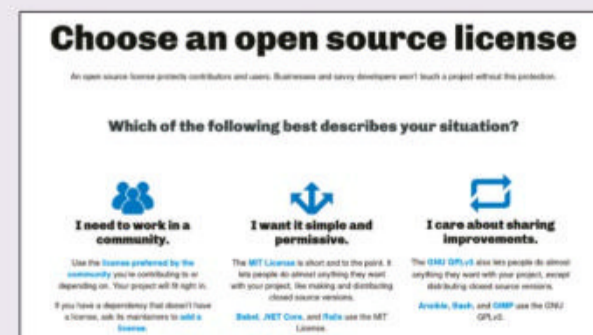
There are probably more open source licences than there are things in Horatio's philosophy, and it wouldn't really be useful to list them in all their

clause-ey glory here. Some, such as the Creative Commons family, have been penned with crafts other than software in mind. There's also the GNU Free Documentation License (GFDL) under so many of those man pages you're supposed to flipping read. We won't even mention the profane WTFPL licence.

But it is important to note that they all arose out of some need, and, as we tell anyone who complains that there are too many Linux distros, if people are willing to take the time to write quality FLOSS then they have every right to release it under whatever terms they like.

The first version of the GPL appeared in 1988, and by 1991 so too had the GPLv2 and the GNU Library GPLv2. The Library licence permitted linking GPL'd

libraries into proprietary executables (under certain conditions) since precluding this would discourage use of those libraries in favour of non-free alternatives. The Library GPL was renamed to the GNU Lesser General Public License in 1999, and is still popular today.



Struggling to find a license? Let GitHub help you. They're owned by Microsoft btw...

for their particular device. But now they can be generated by the free driver itself, making the resulting firmware kosher for kernel inclusion.

There are a couple of GNU-endorsed distros, such as Trisquel, which uses the Linux Libre kernel: a 'de-blobbed' kernel rent asunder from any proprietary bits, whether they be firmware files or microcode or register headers that required proprietary tools to generate/extract. These are great from a software freedom point of view, but those devices that use these de-blobbed drivers may no longer work, or no longer work as well on such distros.

Of the top-tier distros, Debian and Fedora take the strongest stances on software freedom. Software in the Debian repos must conform to the Debian Free Software Guidelines (DFSG), part of the Debian Social

Contract, originally penned in 1997. A number of licences are considered 'DFSG-free', that is, compatible with DFSG's interpretation of software freedom. Likewise, Fedora only accepts software released under certain licences, a long list of which may be found at <https://fedoraproject.org/wiki/Licensing#SoftwareLicenses>. The Open Source Initiative's official definition of Open Source (see <https://opensource.org/docs/osd>) has its roots in the DFSG. And Red Hat's list of approved licences is in part based on those approved by the Free Software Foundation (FSF) and Open Source Initiative (OSI).

DFSG and the Fedora licensing list permit software licensed under any version of the GNU Public License (GPL), any version of the Lesser GNU Public License (LGPL) or the AGPLv3 (which we'll see later has provisions for software running over a network). But free software licences are diverse and manifold, and don't all originate from the GNU project. At least six variations of the BSD License are acceptable for Fedora, with the BSD-3-clause License being the only one listed in the DFSG.

The right to tinker

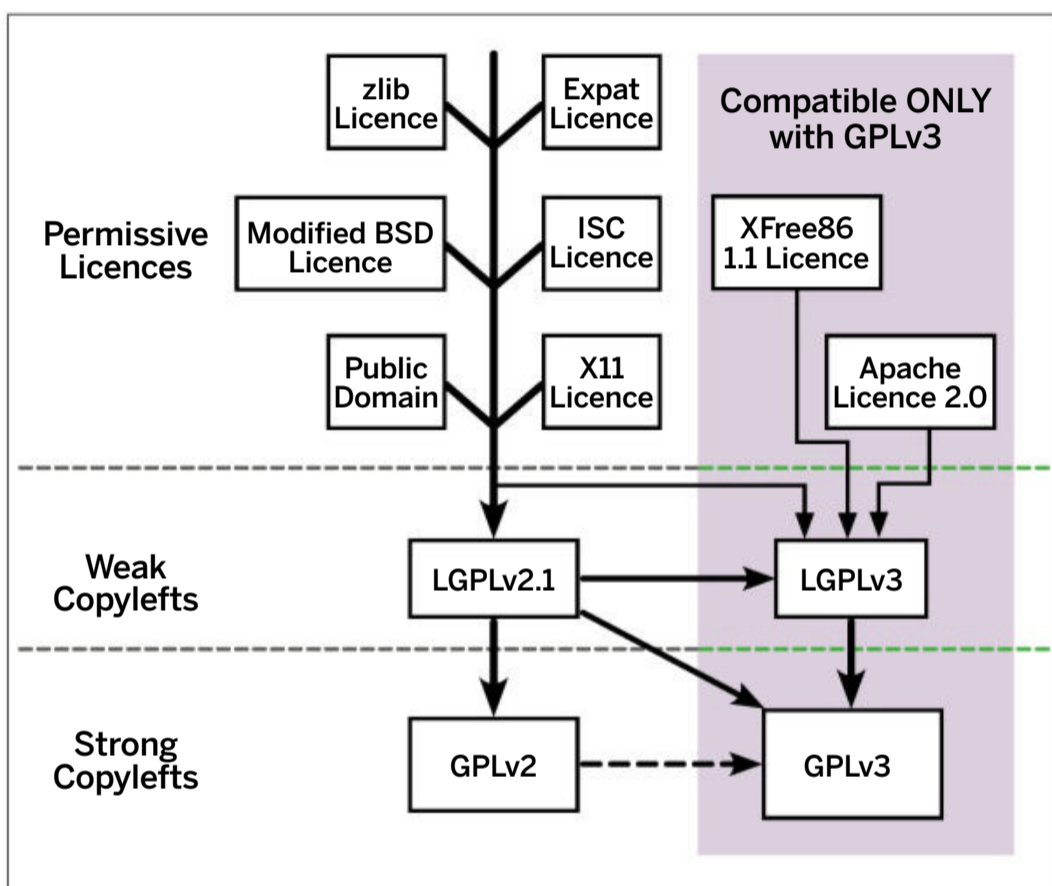
The most popular licence for projects hosted on GitHub is the MIT, which is also the shortest. MIT is a permissive licence which allows users to modify and redistribute (including selling) the software, while indemnifying the authors for any damage arising from use or dealings with the said software.

The key difference between MIT and copyleft-style licences like the GPL is that MIT allows closed source forks and versions. There isn't much substantive difference between the GPLv1 and GPLv2; the latter aimed to clarify some points of common misunderstanding. However it also contained the catchily titled 'Liberty or Death' clause, which forbids distribution of the software in a case where restrictions are imposed (by a court, say) that would prevent user freedoms being respected.

Changes in the way we use software (and the hardware that talks to it) necessitated changes in the licences that accompanied it. The GPL saw its third incarnation in 2007, and took aim at what the FSF saw as three emerging threats: 'Tivo-ization' (where free software is used in devices over which the user has no, or limited, control); laws prohibiting Free Software (software that might break Digital Rights Management and violate the DMCA or EU Copyright directive); and Discriminatory Patent Deals.

These three ideas had no place in a Free Software utopia. The GPLv3 holds sacrosanct the Right to Tinker by requiring distributors of hardware running GPLv3 code to provide instructions for installing modified software. The GPLv3 itself doesn't ban (or even cast in a negative light) DRM, it just says it can't be used to revoke freedoms granted under the GPL.

By the mid-aughts, network services were becoming popular, and the Affero License was introduced so that web applications could be licensed in such a way as to give their users access to their source. But it wasn't without issues. In 2012, Bradley Kuhn – who authored the original Affero clause in the AGPL – pointed out that most uses of AGPL were "shakedown practices". AGPL,



It's fine to relicense code towards copyleft, but you can't take GPL code and release it under a permissive licence. FSF lawyers will hunt you down.

» THE RISE OF AGPL

AGPL is named after Affero Inc, a web services business looking for a licence that would enable other businesses to distribute and modify Affero's code. The resulting licence was written by Affero founder Henry Poole in 2002 and was based on the GPLv2. In 2007 the FSF had released its own version, named the Affero GPL (AGPL). The latest version of the AGPL is essentially the GPLv3 augmented with a new section on Remote Network Interaction, so that these two licences are compatible. Thus GPLed software can be incorporated into AGPL software, and providers of the burgeoning (insert noun here) As A Service industry could help promote and improve free software using the now established copyleft mechanism.

The AGPL took aim at the 'secret sauce/source' model, where Free Software was used in conjunction with proprietary backends. Specifically it targeted the application service provider (ASP) loophole which existed in the GPL, since such providers didn't so much 'distribute' code as let users interact with it while it ran on the provider's hardware.



he said, inadvertently led to a trend of proprietary relicensing. Companies “bait and switched” customers by offering a free product, and then scaring commercial users by spreading fear around the vagaries of free licence conditions into paying for that same product under a different licence. This is what happened to MySQL under Oracle. And this is what almost happened, though for opposite reasons, to Berkeley DB, which Oracle acquired in 2006.

In 2013 it switched the free licence from the historical, GPL-compatible SleepyCat terms to the AGPLv3, as well as continuing to offer commercial licensing. Since AGPL had stronger copyleft provision, there was concern that the industry would shun the database in favour of permissively licensed alternatives.

This all came to a head in 2018 when MongoDB relicensed its eponymous database under a new licence, the Server Side Public License (SSPL). Incidentally, it's perhaps not a coincidence that all the examples of license malfeasance presented here are in connection with databases. There's just a lot of money in databases. Your bank balance lives in a database, for example.

Analyst Billy Duberstein astutely noted in his *If You Invested \$1000 in MongoDB's IPO, This Is How Much Money You'd Have Now* that “if a company wanted to switch vendors, it would have to lift all of that data from the old database and insert it into a new one. That's not only a huge pain; it's also terribly risky, should any data get lost”. So databases have an inherent risk of lock-in, even before licensing comes into it.

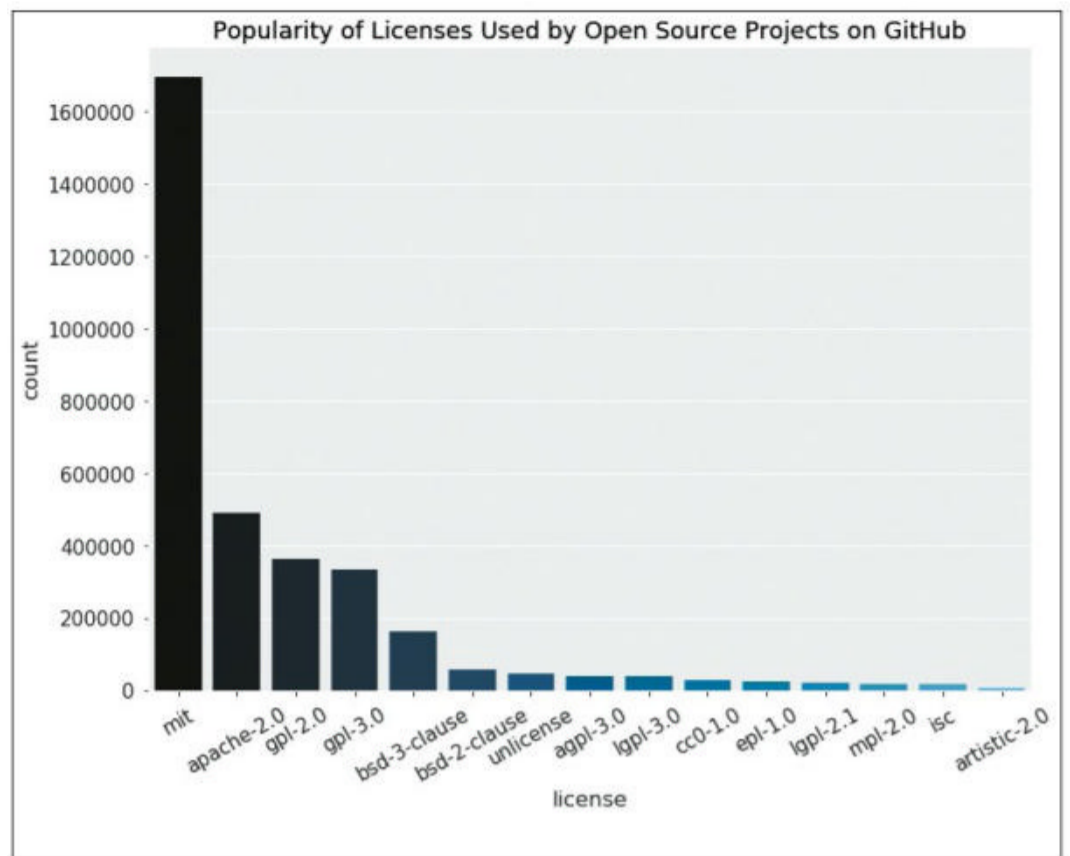
MongoDB's CEO has gone on record as saying that its product was only ever open-sourced as part of a freemium marketing strategy rather than to benefit the community or get help from them. Yet despite this, MongoDB is seen as being more FLOSS-friendly than Oracle. The thing is, a couple of decades before, it was Oracle that was seen as saving the world from expensive IBM databases. This caused sometime LXF-contributor Matt Yonkovit to pen “Is MongoDB the new Oracle?” in 2020, a text which surely you should cast your eyes over at <https://bit.ly/LXF283mongo>.

The SSPL licence stipulates that modifications to software under its conditions must be released (fairly standard copyleft fare), but so too must all surrounding infrastructure. If customers didn't like this, they were free (nay, encouraged) to purchase a commercial licence from MongoDB.

Such a licence wasn't expensive, but what was to stop them pulling an Oracle and jacking up their prices further down the line? “Surrounding infrastructure” is fairly vague, even if you're talking about a simple program, let alone a distributed database behind a pretty web frontend.

Foul deeds

Meanwhile, over at Elastic Inc. a strange new development was brewing. In 2018 a blog post entitled “Doubling down on open” appeared on its website (see <https://www.elastic.co/blog/doubling-down-on-open>) by CEO Shay Banon. The gist of it was that a new Elastic licence was being introduced for parts of their (really, really complicated) database management stack. The post went to pains to point that previously released Apache 2.0-licensed code would remain licensed thusly,



and that (to paraphrase) users should just keep calm and carry on. But something more sinister was afoot; see the box below for just what.

So we've seen that Stallman's trick of inverting copyright to foment innovation in the FOSS ecosystem worked exceptionally well in the GNU era. That success continued well into the web application age, engendering the AGPL to accommodate new ways of interacting with remote code. The GPL enabled companies like Red Hat, Canonical and Docker to thrive. But it also enabled less scrupulous companies to scaremonger customers into paying more, or to harvest all their innovations for themselves. **LXF**

Kaggle analysed 3TB of GitHub repos and confirmed that permissive licences are all the rage nowadays.



» ELASTIC MORALS

MongoDB didn't have to wait long before its SSPL saw some new adoptees. In 2021 Elastic Inc. switched its *ElasticSearch* and *Kibana* products to a dual-licensing arrangement. Non-commercial users would be governed by the Elastic licence (the licence under which it was distributed for the past three years), which outlawed using Elastic in most 'as a service' applications. Commercial users would have to share their surrounding infrastructure (or “service source code” per the license wording).

Previously, *ElasticSearch* was available as an Apache 2.0-licensed distribution, with many of the individual libraries in the Elastic-licensed distribution still governed by that permissive licence. Going forward, there would be no Apache-licensed code and *ElasticSearch* and *Kibana* would “no longer be referred to as open source”.

The main reason given for Elastic's change of tack was to prevent “attempts to splinter our community with 'open' repackaging of our OSS products”, a not particularly subtle dig at AWS's Open Distro for ElasticSearch, which was released in 2019. Open source advocate Vicky M Brasseur described SSPL as “a hostile proprietary license masquerading in open source clothing”.

She goes on: “It's not a stretch to interpret the wording of the license as requiring users of the SSPL'd software therefore to release the code for everything straight down to the bare metal”. We'd encourage you to read Brasseur's post in full at <https://bit.ly/LXF283brass>.

TUTORIALS

CASTERO

Credit: <https://github.com/xgi/castero>

TUI podcast player

Listening to podcasts is as natural to **Shashank Sharma** as making tall claims. He's tried all podcast clients and has finally found one of the best.



OUR EXPERT

Shashank Sharma

is a trial lawyer in Delhi and an avid Arch user. He's always on the hunt for pocket-friendly geeky memorabilia.

Over the past 15 years, podcasts have become an inalienable part of most modern web users' daily lives. Whether it's news or commentary on relevant topics of the day, or informed discussions on niche subjects such as open source software, art of creative writing or comic books, you'll find a range of podcasts on all such subjects and more. Many modern music players now support playing podcasts, but if you're looking for a dedicated podcast client for the terminal, *Castero* just might fit your needs.

The old Unix philosophy of having programs that do one thing well is all but forgotten, with behemoth graphical applications that strive to do a variety of different tasks. For instance, consider your web browser, which can display web pages, but also play multimedia files, act as a download or FTP manager, screencaster and more.

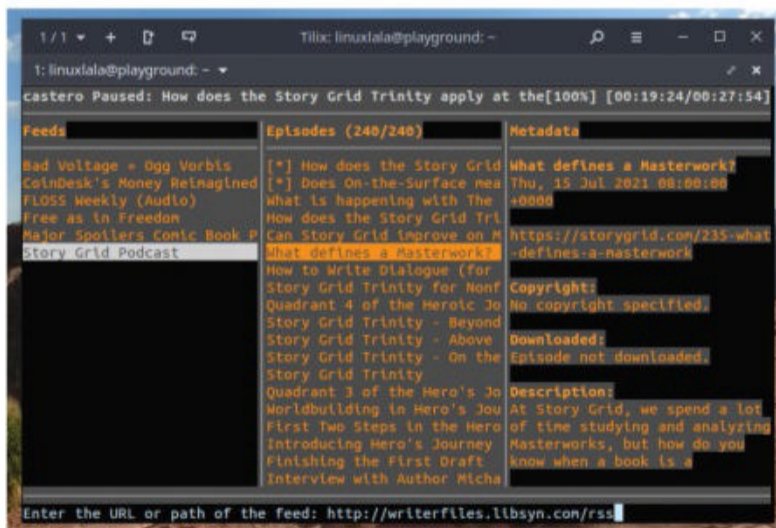
Thankfully, the old Unix philosophy lives on in many text-based applications. *Castero*, a TUI podcast player, is one such application. It's unable to double up as a music player, and it can't be used as a web browser. Its only purpose is to be an efficient podcast client, and nothing more. To that end, the program supports a variety of useful features such as adding RSS feeds, fetching episodes and metadata information, downloading episodes for offline listening and increasing/decreasing playback speeds.

First steps

You won't find *Castero* in the software repositories of many distributions, but installation is still fairly straightforward because the project doesn't have complex dependencies. Released under the MIT licence, the project only requires SQLite3, Python and a choice of either *MPV* or *VLC* music players. Each of these can be installed using the software repositories on most desktop distributions such as Arch, Mageia, Debian, Fedora, Ubuntu and Mint.

If you already have all the dependencies and *pip* installed on your distribution, you can install *Castero* with the `pip3 install castero` command. You can now launch *Castero* by running the `castero` command. The project defaults to a three-pane interface.

To add a new podcast feed to *Castero*, press 'a'. The bottom of the interface will then change to enable you to type or paste the URL of the RSS feed for the podcast you wish to listen to. Press Enter and *Castero* will then fetch the list of episodes for the specified Podcast feed.



Castero refers to the different panes as menus. Unfortunately, it's not possible to change the width of any of the panes/menus.

» FILTERING CONTENT

You can enable the filter by pressing `/`. The feature works across the selected panes. So, if you're in the Feeds pane then it'll show all podcast feeds that match the specified keyword. Similarly, when you're in the Episodes pane, the filter will display all episodes that match the keyword.

The feature works flawlessly when filtering episodes. So, if you were to search for 'crazy' in the Episodes pane, *Castero* will display all episodes that

feature 'crazy' in the title for the currently select podcast feed. However, when you switch to a different podcast feed, the episodes will still be restricted to ones that feature 'crazy' in the title. To exit the filter, you must press `/` again. Now *Castero* will show all episodes for each of the different podcast feeds.

You can similarly filter content in the Feeds pane, which already displays an alphabetical list of added feeds. So, if you have Bad Voltage, CoinDesk, Fedora,

Story Grid, Ubuntu and other podcast feeds in the list, but want to search for 'grid', the layout will show Bad Voltage in the Feeds pane, but all episodes from the Story Grid podcast in the Episodes pane. That is, the first podcast in the Feeds list is displayed irrespective of the search term or matches found, but the Episodes list displays the correct content.

Furthermore, the filter feature isn't case sensitive, so searching for 'coin' will match 'coin' as well as 'Coin', and so on.

```

1 / 1 + ... linuxlala@playground: ~/config/castero
1: linuxlala@playground: ~/config/castero
GNU nano 4.8 castero.conf
[playback]
# The distance to move forward when pressing seek keys, in seconds.
# default: 30
seek_distance_forward = 30

# The distance to move backward when pressing seek keys, in seconds.
# default: 10
seek_distance_backward = 10

# The default playback speed. See also the rate_increase/rate_decrease keys.
# default: 1.0
default_playback_speed = 1.0

# The default volume. Can be 0-100, inclusive.
# default: 100
default_volume = 100

# The amount to adjust the volume when pressing the volume up/down keys.
# default: 5
volume_adjust_distance = 5

Get Help Write Out Where Is Cut Text Justify Cur Pos
Exit Read File Replace Paste Text To Spell Go To Line

```

You can change the default keybindings for all commands, such as seek help, seek forward/backward and add/delete feed, from under the [keys] heading of the config file.

You can similarly add feeds of the all the podcasts you're interested in. Unfortunately, *Castero* doesn't make it possible to sort the podcasts under the Feeds pane, and all the podcasts are listed alphabetically.

The Episodes pane in the middle displays all the episodes for the selected podcast, while the third pane provides metadata information such as the name of the episode, release date and license information, if any.

Note that adding a feed means that *Castero* will download only a list of all available episodes, but not the episodes themselves. This means that having a dozen or more feeds, with thousands of available episodes doesn't take up any disk space on your machine. It's only when you choose to download episodes for offline listening that they'll take up any disk space at all.

Listening to podcasts

Unlike graphical applications that can be controlled using a mouse, or trackpad, many CLI utilities can only be controlled using the keyboard. With *Castero*, you can use the scroll wheel on your mouse to scroll through the Feeds and Episodes pane, but you can't left-click to switch between the different panes, or even select a feed or episode. Press 'h' for a list of all available key commands you can use to control *Castero*.

Once you've added several podcasts, you can navigate through the list of podcasts on Feeds pane using the up/down arrow keys. You can similarly use the right/left arrow keys to move between the Feeds and the Episodes panes on the interface.

To listen to an episode of a podcast, select it in the Episodes pane, and then press Enter. The top-right of the interface shows the duration and tracks the progress of the episode as it's played. You can use the 'f' and 'b' keys to fetch forward and backwards during playback. You can similarly use the ']' and '[' keys to increase or decrease the playback speeds. *Castero* remembers the location where you pause an episode, so you can restart it from the same point later.

You can also add several episodes from the same or different feeds to the playing queue. Select the episode in the Episodes pane and then press Space. Next, select the next episode and press Space again. Repeat the process to add multiple episodes to the queue.

The default *Castero* interface doesn't show a list of all the episodes in the playing queue. You can switch between layouts using the '1', '2', '3' and '4' keys.

The '2' interface shows the episodes in queue. You can pause or play episodes from the queue by pressing 'p'. Unfortunately, working with queues is somewhat erratic, and *Castero* stopped responding, and on occasion deleted episodes from the queue without any reason.

Keybinding	Layout description
1	Default three-pane layout featuring Feeds, Episodes and Metadata panes.
2	A two-pane layout that shows episodes in the queue.
3	A two-pane layout featuring Feeds and Episodes.
4	A two-pane layout featuring all downloaded episodes and metadata.
5	A two-pane layout featuring all episodes across all feeds sorted by release date.

Configuring Castero

You can edit the keybindings, default colour scheme, define the location to download podcasts and more by editing the configuration file. It's best to preserve a copy of the default configuration before tinkering with it.

Open the terminal and run the `cp ~/.config/castero/castero.conf ~/.config/castero/castero.conf-orig` command. This will create a copy of the default configuration file so that you can revert to it if needed.

You can now open the `~/.config/castero/castero.conf` in your favourite editor. The config file is well commented, so it's quite easy to navigate.

Under the [client] heading you'll find all the configurable elements that define *Castero*'s appearance and behaviour. For instance, whether or not to display vertical lines to separate the different panes in a layout, or whether *Castero* should ask for confirmation when you press 'd' to delete the selected feed.

You can similarly define the default downloads directory for episodes under the [downloads] heading. By default, episodes are downloaded in the `~/.local/share/castero/downloaded` directory. Navigate to the `custom_download_dir =` line and add the path you wish to use instead.

Castero supports a number of different colours, and they can be used to liven up the interface. Although the default colour scheme is pleasant enough, you can also make changes to different elements listed under the [colors] heading.

By default, *Castero* jumps forward by 30 seconds and backwards by 10, but you can change this too from under the [playback] heading.

When you make any changes to the config file, you must save the file and restart *Castero* for changes to take effect.

Castero is one of the simplest projects we've come across, and it's also enjoyable to use and make your own. With a host of useful features, *Castero* is also likely to become your default podcast client. **LXF**

QUICK TIP

You can use the Issues section of the project's GitHub page (<https://github.com/xgi/castero>) to report erratic behaviour or even request new features. Make sure to scroll through the closed issues to see if the problem you're facing has already been addressed. You can similarly use the search feature to ensure that a feature you might want hasn't already been requested.

PDF EDITORS

Credit: <https://okular.kde.org>

The complete guide to editing PDF files

Nick Peers reveals all the free tools you need to view, annotate, edit and create PDFs on your Linux-running PC.



OUR EXPERT

Nick Peers loves the PDF format. How else do you explain the thousands of PDF documents scattered across his hard drives?

The humble Portable Document Format file (PDF to its mates) is one of the most versatile formats you could ask for. It solves a very simple problem: how do you ensure that a file you share with other people looks identical on their computers even if they don't have access to the application in which it was created?

After starting off as a simple way of sharing and displaying documents in a consistent way, PDFs have evolved to become ever more useful. Armed with the right tools you can search them for specific text (then extract that text), annotate them, add comments and even use them like digital forms, filling in sections by typing text or selecting from drop-down menus. And if the PDF is a 'true' one (see the box right), you can even

edit its content and layout. In this tutorial we'll reveal what free PDF tools are available in Linux, plus how to use them to meet your every PDF-related need.

Creating PDFs

Before we delve any deeper, let's quickly recap how you can create PDF files of your own. The easiest way to generate them is from existing documents. In *LibreOffice*, for example, choose 'File > Export as > Export as PDF' to generate a PDF from your *Writer*, *Calc*, *Impress* or *Draw* document. You'll see lots of different options are available, which you'll come to understand better later on in this tutorial – for now, in most cases the default settings are fine, so simply click Export.

Other programs – if they support PDF directly – should work in a similar way, so look under the Export or 'Save as' menus for options. If none exist, try File > Print instead. Select the 'Print to File' option and make sure the output format is set to PDF. You'll be able to generate a PDF from the entire document or by selecting specific pages.

Viewing PDFs

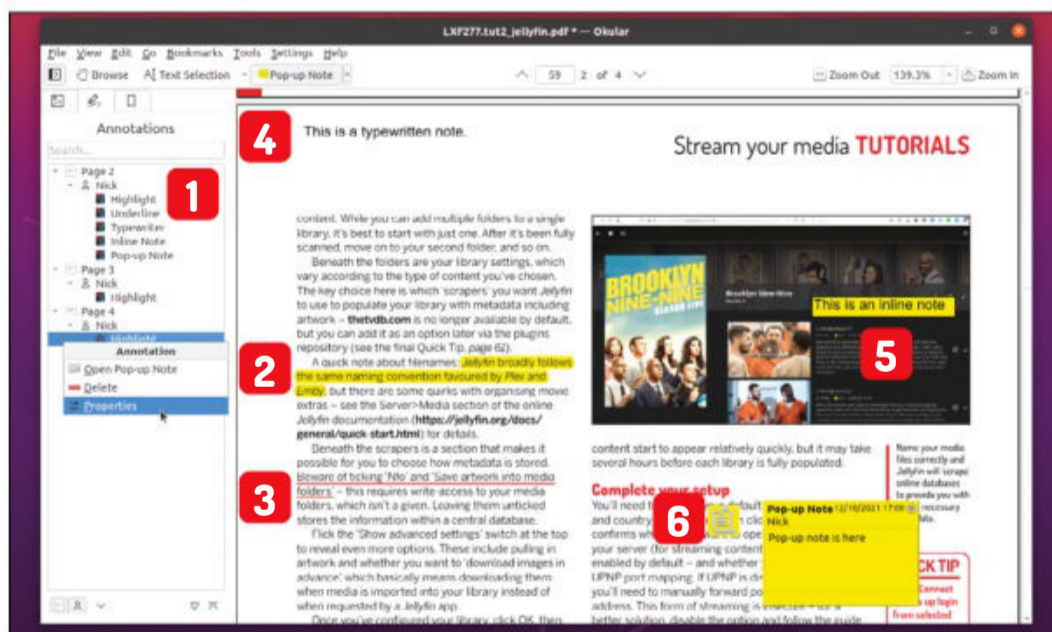
Most distros ship with tools for viewing PDFs (along with other files) – in the case of Ubuntu that tool is *evince*, better known as *Document Viewer*. It's pretty basic but does provide all the core functionality you need for viewing PDFs. Simply double-click a PDF file, and it'll open in *Document Viewer*.

Once opened, if the file has an editable text layer, you can easily search for, select and copy text to the clipboard for pasting elsewhere, say into your text editor. You can also extract images from the PDF – simply right-click on an image in the document and choose 'Save image as...'

Document Viewer can also fill in certain PDF forms – basically any except those created in Adobe's XFA format (see the box on page 64). You can also rotate individual pages of a PDF to switch orientation between portrait and landscape – you'll find this option under the 'hamburger' (☰) button, which opens *Document Viewer*'s main menu.

Document Viewer is adequate for basic PDF usage, but if you want to go further with your PDFs then start by installing a more powerful tool in the form of *Okular*.

LABEL PDFS WITH OKULAR



1 Annotations list
The Annotations lists all the labels on the document. Right-click one to select it, delete it or tweak its properties.

2 Highlighter
Both yellow and green highlighters are available (other colours can be configured). Click and drag across a passage of text.

3 Underline
If you prefer to underline content, use this tool. It's red by default, but you can choose a different hue and level of opacity.

4 Insert Text
This option enables you to type – like a typewriter – on top of the document itself. You can change font, size and colour.

5 Inline notes
These notes appear like sticky notes – just click where you want one to appear and start typing.

6 Pop-up notes
Instead of the full text appearing on top of the document – as with an inline note – Pop-ups display this icon instead.



Ubuntu's built-in Document Viewer is a capable – if basic – PDF viewing tool. Its only editing capabilities are a page rotation tool.

It's a universal document viewer with more flexible viewing and text selection options, plus it enables you to annotate PDFs in several different ways.

Focus on Okular

Okular is available through Snap and the *Software Centre*, but if you've installed Flatpak you can install the very latest version directly from the command line:

```
$ flatpak install flathub org.kde.okular
```

Once installed, *Okular* can be opened via the Launcher. You'll immediately see its user interface is more feature-packed than *Document Viewer*, so let's see what additional tools it offers for interacting with PDFs.

First, it offers additional ways of selecting content: in addition to the basic text selection tool (click and drag to select one or more lines of text), there are area and table selection tools. The area selection tool is particularly powerful: click and drag a rectangle around the area of the page you wish to capture, and you'll be rewarded with a pop-up menu.

This is divided into two sections: text and images. The former allows you to copy any text within the selection to the clipboard, have it spoken out loud, or used as the basis of a document or web search. The image section allows you to either copy the selection to the clipboard or save it as a file. The table selection tool is designed to make it easy to extract columns and rows of data.

Annotate PDFs

When reading paper documents, there may be times when you'd like to record some notes – typically in the margin – or highlight text. Maybe you'd like to insert a bookmark to jump back to a key passage at a later date. Thanks to *Okular*, you can take these tools into the

digital sphere too. The aptly titled annotation opposite reveals how each different tool works, and you'll find six of the most popular are accessible from the Quick Annotation Tools pop-up menu on the main toolbar.

They're all pretty self-explanatory to use – *Okular* supports undo (Ctrl+Z) if you'd like to experiment, but you can also access additional tools by clicking the drop-down menu and choosing 'Show more annotation tools' to bring up a separate toolbar. Alternatively, add additional annotation tools to the Quick Annotation Tools menu by selecting 'Configure Annotations'. Click Add to set up a new tool – you can create variations on existing tools (say a different coloured inline note) or set up additional types.

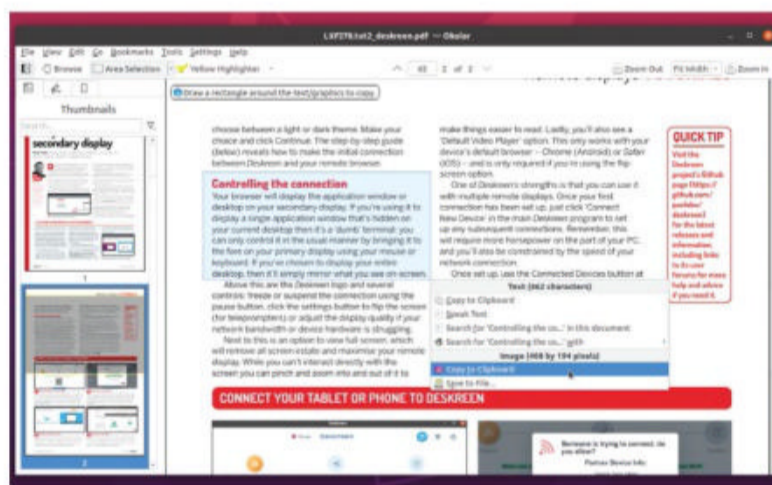
These extra types include lines (freehand or straight), shapes (geometrical or freehand polygon) and stamps. The latter is interesting as it enables you to – for example – 'stamp' the document with phrases like 'for public release' or 'approved', but at present those stamps won't appear to other users unless they open the PDF in their own copy of *Okular*.

One final annotation tip: you can hide all annotations from view for a clear view of the document without removing them by pressing F6. Press the key again to bring them back.

More viewing aids

Away from annotating documents, *Okular* also supports bookmarks. These enable you to bookmark individual pages to jump between. Simply navigate to the page in question and choose 'Bookmarks > Add Bookmark' (or press Ctrl+B).

Once added, bookmark icons appear on top of the page thumbnails in the left-hand navigation page, but you can also view all bookmarks via their own dedicated Bookmarks tab from here too. By default, each



QUICK TIP

Looking for a way to edit PDFs without having to pay for the privilege? If you don't have access to the original document and its parent application, open the PDF in LibreOffice Draw. You can swap graphics, make changes to the text and more besides.

Okular's rectangular selection tool allows you to scoop up text and/or images from the document via a handy pop-up menu.

» TYPES OF PDF

There are, broadly speaking, two types of PDF: 'true' or digitally-created PDFs, and 'image-only' or scanned PDFs. When you snap a document using a digital camera or scanner, you effectively take an image of that document, one image per page snapped. Each image can be viewed by the naked eye, but any text on that image is inaccessible to you unless you perform OCR (Optical Character Recognition) on the document using specialised software,

which basically scans the image for recognisable text. It then places this text onto a separate – but invisible – layer that allows you to search for, select and extract text from the document.

A 'true' PDF is one that's created within a program – either via its Export or 'Save as' menu, or by printing to a PDF virtual printer. This creates separate text and image layers, each of which can be accessed via the right PDF tools, such as

those featured in this tutorial. One variation of a 'true' PDF is one that's been generated to allow users to fill in forms by typing in details into fields on the document or by making selections using options such as tick boxes, radio buttons and drop-down menus.

Note, however, that not all forms are equal – many use Adobe's XFA format, which won't work in *Okular* or Ubuntu's built-in *Document Viewer*.



QUICK TIP

If you'd like to make Okular your default PDF viewing tool in Ubuntu, right-click any PDF file and choose Properties. Switch to the Open With tab, select Okular from the list and click 'Set as default'.

bookmarks is named after its page reference, but right-click an entry under Bookmarks and you can rename it to something more descriptive. In terms of navigating between bookmarks in a document, use the Bookmarks menu where you'll see all bookmarks conveniently listed. You'll also see options for removing or renaming the current page's bookmark if it exists.

One final handy tool that *Okular* offers is a magnifying glass. If for any reason your PDF contains microscopic text, select Tools > Magnifier. You'll see a crosshairs pointer appear – click and hold on the page to bring up a magnification tool that will make the text more readable or images easier to make out. Sadly you can't adjust the magnification level from the default 10x – something for a future update, perhaps?

Create a digital signature

Because they deal with documents, PDF files may occasionally require some form of validation to prove they're from a specific source. This comes in the form of a digital signature, which requires you first set up a digital certificate you can use to prove you're who you say you are.

The simplest way to sign PDF documents is through the web using a tool like eSign PDF (<https://smallpdf.com/sign-pdf>). But if you've already got a digital certificate installed, you can use it to sign PDFs locally too. *Okular* can technically do this, but it requires you

install version 21.01 or later of the Poppler PDF rendering library (see <https://poppler.freedesktop.org>), which basically means compiling and installing the latest version manually if you're running Ubuntu 20.04.

Instead, you can sign PDFs using *LibreOffice Draw* – choose 'File > Digital Signatures > Sign Existing PDF...' After loading your file, you'll see *LibreOffice Draw* has opened it read-only, so click 'Sign Document' at the top. You'll be prompted to choose your digital signature from the list – select it and click 'Sign Document...' to do so.

Doing more with PDFs

So far we've focused on viewing and labelling PDFs, but what if you want to make changes to the page layout of a PDF? For example, you might want to delete unwanted pages from a document, change the page running order or even combine two or more PDFs into a single file.

The solution lies with two tools. The first is *PDF Mix Tool* (<https://scarpetta.eu/pdfmixtool>), which offers you nine ways to manage PDF files. Again, while it can be installed via Snap and the *Software Centre*, the Flatpak build is the newest available release:

```
$ flatpak install flathub eu.scarpetta.PDFMixTool
```

Once installed, open *PDF Mix Tool* via the Launcher. It's a simple two-paneled window, with available options clearly laid out on the left. Most are self-explanatory – Merge PDF Files allows you to combine multiple PDF files into a single document. Simply click 'Add PDF file' to select your files in any order, then use the 'Move up' and 'Move down' buttons to arrange them in the order you wish them to appear in the document and finally click 'Generate PDF'.

If you'd like to mix things up a little, choose 'Alternate mix' instead. You add and arrange files in exactly the same way, but on clicking 'Generate PDF' *PDF Mix Tool* will extract the first page of each document in turn to add to your single file, then move on to the second page of each document, then the third page and so on until all the pages have been merged into a single PDF.

Work with single files

The rest of *PDF Mix Tool*'s feature set works with single files only. 'Extract pages' allows you to extract individual or groups of pages from your document, then either output them as individual files for each page, or as a single, truncated document. Select 'Pages' to select your pages using the following syntax: 1-3,4,7,8-10.

'Delete pages' allows you to remove pages from your document, then either save over the original or choose 'Save as...' to create a copy. It follows the same page-selection syntax as the 'Extract pages' option.

'Rotate' works in a similar way to *Document Viewer*'s rotate tool, except it's far more versatile: first you can set a specific 90-degree rotation (90, 180 or 270), plus you can apply the change to multiple pages of your choosing. And again, you get to choose between overwriting the original with your changes or creating a fresh copy – this feature is of most use when trying to change the orientation of pages from portrait to landscape (and vice versa).

The 'Pages Layouts' section is particularly interesting. It enables you to change the size, orientation, alignment and margins of each page, but in addition you can also compress multiple pages into a single page (similar to how you'd create a photo



The major problem with Okular's magnifier is that it's set to 10x magnification with no means of adjusting.

» FILL IN ADOBE FORMS

Not all PDF forms are created equally. Those that adhere to Adobe's XFA standards can't be filled using either *Document Viewer*, which simply won't allow you to edit the fields, or *Okular*, which will at least warn you that it can't be used to fill in the form.

If you've received an XFA-formatted PDF that needs filling, then you'll need to install the free version of *Master PDF Editor*, which you can download and install as a DEB file via its website (<https://code-industry.net/free-pdf-editor>). Once installed, simply open the file in *Master PDF Editor*, and you should find the form can now be filled.

The free version of the program also offers some other handy features: you can create PDF documents from scratch – either by importing compatible image files, or from paper using your scanner. Other features supported without a registration code include splitting, merging, commenting and annotating, plus bookmark management.

The full version adds full editing capabilities as well as support for digitally signing your PDFs. You can road-test all these features within the free version – *Master PDF Editor* will embed a watermark into any documents saved using the unregistered version. If you like what you see, registration costs £64.50.

thumbnail sheet) – the step-by-step guide on the opposite page reveals how it works.

If you'd like to make changes to a document's metadata – its title, author, subject, keywords and so on – then navigate to 'Document Information'. Again, don't forget to click Save or Save as... when you're done to record your edits.

'Booklet' is designed to allow you to convert a PDF document into one optimised for printing double-sided and folding in half. The resulting PDF will look nonsensical in your viewer, but when you print it out and fold it, you should find it creates the perfect half-fold documents, ready for stapling in the middle, saddle-stitch style.

Last, but not least, 'Add empty pages' does what it says on the tin: open your file, choose how many blank pages to add, whether to make the same size as the rest of the document (or specify your own size and orientation), and finally where to place those empty pages within the document (before or after a specified page, for example).

Rearrange the order of pages

PDF Mix Tool is packed with useful features, but it has one glaring omission: the ability to easily move pages around within a single document. Thankfully that gap is filled by the aptly titled *PDF Arranger* (<https://github.com/pdfarranger/pdfarranger>). It offers a similar set of features to *PDF Mix Tool* but wraps them up in a very different user interface. Once again, the Flatpak build is the latest version:

```
$ flatpak install flathub com.github.jeromerobert.pdfarranger
```

PDF Arranger's UI is clearly designed with one principal aim in mind: to make it easier to rearrange pages within a document using drag-and-drop. Start with the button in the top left corner to add your first document. You'll see it's displayed in a grid-like



Handy thumbnails of each page makes it easy to see what's where in PDF Arranger. You can then rearrange pages using drag and drop.

thumbnail view, making it easy to identify individual pages within the document.

From here, you can add more documents to the mix, combining them all into a single document like *PDF Mix Tool's* 'Merge PDF Files' feature, but here's the clincher. Once all your pages are in place, you can simply drag and drop them around the main window to change their running order.

To move groups of pages together, simply Ctrl-click to select each page in turn (or Shift-click to grab consecutive pages). Once highlighted, you can drag these to their new location, or right-click for more options.

The right-click menu reveals a host of tools, from deleting and rotating pages to changing the page format (basically scaling or cropping margins). You can duplicate pages, split them, reverse their order or export the selection as a new PDF. Most of *PDF Mix Tool's* functionality is duplicated here, but there are enough omissions – specifically the page layout tools – to ensure both apps serve more as friendly companions than competition. **LXF**

QUICK TIP

If you're happy to use the command line, you can convert image-only PDFs into files with an editable text layer using **OCRmyPDF**. Visit <https://github.com/jbarlow83/OCRmyPDF> for details of how to install and use it, plus what additional dependencies you'll need.

CREATE A PDF THUMBNAİL SHEET



1 Set up thumbnail sheet Open *PDF Mix Tool*, select 'Pages layout' and click 'Open PDF file' to select the file you wish to display on your thumbnail sheet. Once imported, use the 'Page size' fields to select the size (and orientation) of the sheet itself. If you plan to print the sheet, then A4 or – if your printer can handle it – A3 is best.



2 Choose thumbnail layouts Use the 'Pages layout' to set how many pages will be displayed on each individual page of the sheet – use the Rows and Columns fields as shown (for A4 portrait, 2x4 or 3x6 is a good choice; for landscape, reverse it to 4x2 or 6x3). Use the Spacing field under 'Pages Layout' in conjunction with the Margins settings to create spacing around each page.



3 Save and review Click 'Save as...' to create your PDF thumbnail sheet without overwriting the original document. Once done, open it in your PDF viewer tool to review how it looks – you'll see pages are listed sequentially left to right along each row, enabling you to see the document's overall layout. If necessary, you can return to *PDF Mix Tool* to make further tweaks.

» GET ACCESS TO ALL OF OUR PDFS Subscribe now at <http://bit.ly/LinuxFormat>

BACK ISSUES » MISSED ONE?

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 five 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

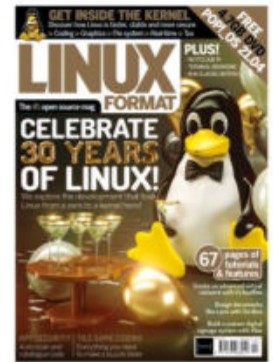
Product code:
LXFDB0280

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).



ISSUE 278
August 2021

Product code:
LXFDB0278

In the magazine

Want a faster, better server? We show you how to set up some of the best server distros around. We also cover desktop virtualisation, running a mobile second screen, emulating the Altair 8800, multitasking in Python and web-app security.

DVD highlights

Bodhi 6.0 and Lubuntu 21.04 (both 64-bit) and AntiX 19.4 (32-bit).



ISSUE 277
July 2021

Product code:
LXFDB0277

In the magazine

Discover what's new in the latest version of Ubuntu, grab a slice of network-attached storage, code a game in Scratch, emulate the Dragon 32 and set up your own streaming server with Jellyfin. Plus we look back at Prestel, the pre-internet data service!

DVD highlights

Ubuntu 21.04 (64-bit) and MX Linux 19.4 (32-bit).



To order, visit www.magazinesdirect.com

Select Single Issues from the tab menu, then select Linux Format.

Or call the back issues hotline on **0330 333 1113**

or **+44 (0)330 333 1113** for overseas orders.

Quote the Product code shown above and have your credit or debit card details ready

READING IN THE USA?

UK readers
turn to
p12

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!



» USA
From \$132
For 13 issues

» REST OF THE WORLD
From \$137
For 13 issues

» EUROPE
From €100
For 13 issues

IT'S EASY TO SUBSCRIBE!

Visit www.magazinesdirect.com/linux-format

Call +44 0330 333 1113

Lines open Monday-Friday, 9am-5pm, UK time

EMULATION

Babbage and running his many engines

Mike Bedford shows you how to program the mechanical computing devices that pre-dated modern electronic computers by over a century.



OUR EXPERT

Mike Bedford Like most of us, Mike tended to think of the likes of Alan Turing and John von Neumann as the pioneers of computing, so delving further into Charles Babbage's creations was truly inspiring.

QUICK TIP

The working Difference Engine is on public display at the London Science Museum so you can see Babbage's creation in all its glory. Also on display is a small portion of the Analytical Engine's mill that was built by Charles Babbage.

Discussion about the first ever computer often results in heated debate fuelled, no doubt, by a degree of patriotism, but also by the vagaries of what's meant by 'the first computer'. More precise terms help us get closer to the facts. There's widespread agreement, for example, that the first programmable, digital electronic computer was Colossus, as designed by British codebreakers in 1943 at the Government Code and Cypher School at Bletchley Park. But it differed from today's computers in being programmed by flipping switches and plugging patch leads and, most importantly, because it was designed with particular jobs in mind, it wasn't a general purpose machine.

Two years later, the University of Pennsylvania completed a massive machine containing thousands of thermionic valves – or electron tubes if you prefer – in a project funded by the US Army. It was called ENIAC – which stood for Electronic Numerical Integrator and Computer – and unlike Colossus, it was Turing-complete, which meant it was a universal computer. It's generally recognised as the first electronic programmable computer. Like Colossus, though, it didn't hold its program in memory, so programming was a laborious process of patching leads and switches.

Next up is the machine that I'm most comfortable about calling the first ever computer, although admittedly that's in no small part because it was built less than 50 miles from the place I call home. This was Manchester University's modestly named SSEM (Small-Scale Experimental Machine), or colloquially the Manchester Baby. It ran its first program in 1948, it was a universal computer, and it held its program in memory, just like all today's mainstream machines.

Bizarrely, though, computers don't have to be fuelled by electronic circuits. Today, researchers are delving into biological computers, optical devices and machines powered by chemical reactions, to name but a few, but our subject here is a concept of computing that takes us back to the 19th century when advanced technology utilised cogs, gears, levers and steam engines. This is our earliest foray into reliving historical computers, it features the work of Charles Babbage, and this isn't just a history lesson. If you want some hands-on experience, we'll show you how to use a spreadsheet and an



Charles Babbage has been called the Father of Computing, even though he never completed any of his groundbreaking designs.

CREDIT: Public Domain, <https://www.wikidata.org/wiki/Q52156071>

emulator to exercise a couple of computing devices, one of which was designed 200 years ago.

The Difference Engine

Charles Babbage was born in London on 26 December 1791. He studied mathematics at Cambridge, but wasn't a star student. His early career wasn't too inspiring either, despite making a contribution to electromagnetic theory, and dabbling in politics. However, his role as a founding member of the Royal Astronomical Society proved to be key to his lasting legacy. Together with astronomer John Herschel, Babbage had been tasked by the Society with leading a project to recalculate the tables of lunar distances, which appeared in the Nautical Almanac, and which were used in calculating longitude at sea.

This work revealed several errors in those tables and led to Babbage's interest in creating a device to mechanise the calculation of such tables. This culminated in his design of the Difference Engine, although sadly he never got to see his creation in operation. In fact, it didn't see the light of day until 1991,

when it was built by the London Science Museum, working from Babbage's plans. It worked just as its inventor had intended all those years before.

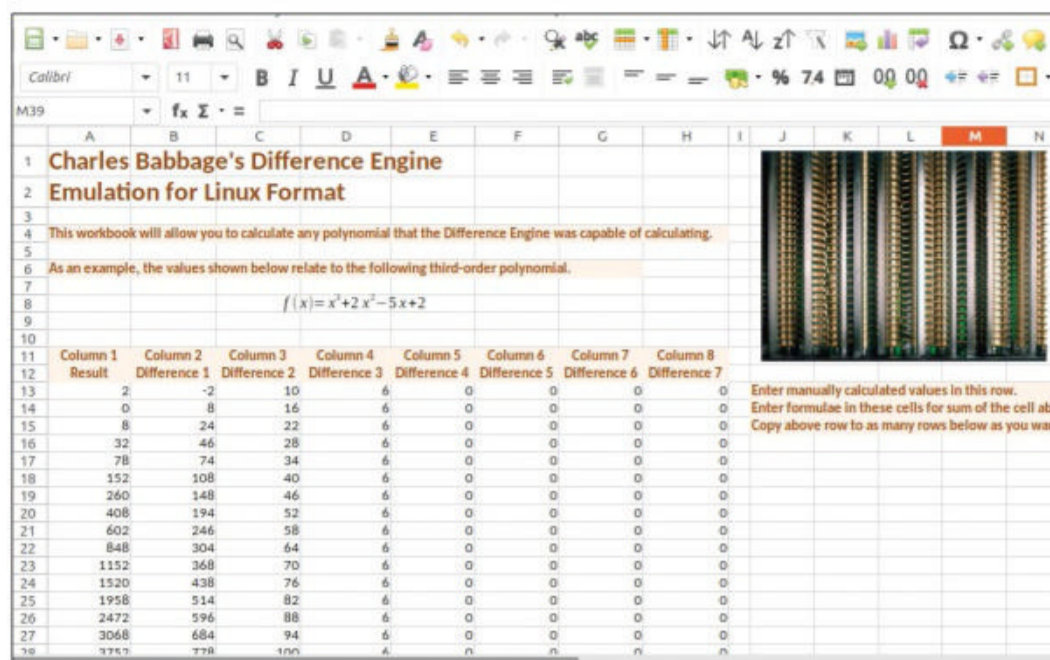
Here we have to get back into semantics because, although the definition of the first computer is open to interpretation, it's a commonly held view that the Difference Engine wasn't really a computer. Instead, it's suggested, it's a calculator. To be honest, we don't really care what it should be called, but we'll start our investigations of Babbage's work here, because it led the way to his Analytical Engine that most definitely was a computer, and a general-purpose one at that.

The Difference Engine was used for calculating tables of polynomials. If recollections of that term from school maths lessons are vague, a polynomial is a function of x that has the form $a_n x^n + \dots + a_3 x^3 + a_2 x^2 + a_1 x + a_0$, where n is the order of the polynomial and the a_n values are constants. So, for example, $x^3 + 2x^2 - 5x + 2$ is an example of a third order polynomial. Polynomials are important because many functions, including logarithmic and trigonometric functions, can be approximated using polynomials.

While the computation of a polynomial isn't difficult, it's easy to see how error-prone it would have been when calculating with just paper and pen, and why Babbage was convinced there should be a better way. Babbage's solution employed the method of differences and relied on an interesting property of polynomials. Take a look at the following table that relates to our previous example of a third-order polynomial.

x	f(x)	Diff 1	Diff 2	Diff 3
0	2	-2	10	6
1	0	8	16	6
2	8	24	22	6
3	32	46	28	6

The first two columns are values of x and the polynomial value for x , and the next three columns are differences. Specifically, Diff 1 contains differences between successive values of $f(x)$, Diff 2 contains differences between successive values of Diff 1, and Diff 3 contains differences between successive values of Diff 2.



Although its mechanical embodiment was a major feat of Victorian engineering, the principles of the Difference Engine can easily be demonstrated using a spreadsheet like Calc.

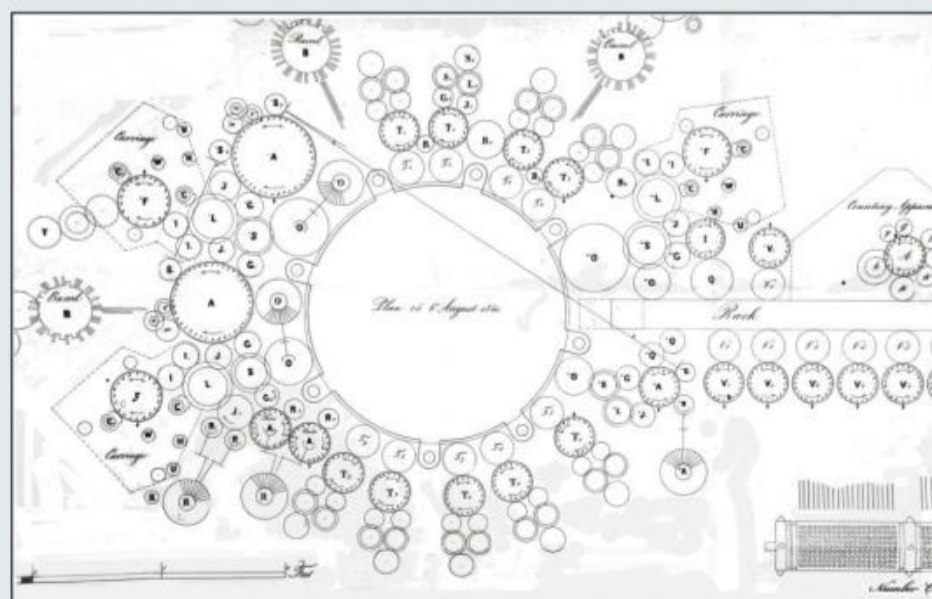
2. You'll see that all the Diff 3 values are the same, and this will always be the case of a third order polynomial.

Similarly, Diff 4 values will always be the same for a fourth order differential and so forth. This provides a method of calculating polynomial values that only involves addition, and which you can easily try out in *LibreOffice Calc*, or your favourite spreadsheet. The information needed for the automated calculation, that would have been calculated by hand, are those shown in the top row of the table, although to get that you'd have to work out values for larger values of x . Having entered those values at the top of five columns in the spreadsheet, it should be fairly simple to figure out the additions that are needed in the rows below or, if all else fails, take a look at the screenshot above.

Following on from this simple example, you might also like to try an exercise which gives a better feel for the value of polynomial approximations. The well-known exponential mathematical function, which has the formula e^x , and which is represented in *Calc* by the `exp()` function, is approximated for small values of x by the polynomial $x^5/120 + x^4/24 + x^3/6 + x^2/2 + x + 1$. We suggest that you create a spreadsheet to evaluate the

» PLAN 28

The London Science Museum's building of a Difference Engine to Babbage's plans might have been a remarkable achievement, but British software engineer John Graham-Cumming plans to go one better by building an Analytical Engine. You can catch up on the latest news of the project at plan28.org. First announced in 2010, three phases were envisaged. First it was planned to study Babbage's design to understand the term Analytical Engine, and which of his many designs should be built. Second, the Engine would be emulated on a modern-day computer to check it out and find any errors in the plans. This would also lead to a CAD exercise in designing each of the parts. And finally, a real Analytical Engine would be built. It's been suggested that it will be five metres tall by eight metres long, making it about the size of a small locomotive. Unlike the Difference Engine, it couldn't have been powered by manual labour, so its motive force would come from a steam engine.



The Plan 28 project aims to bring Babbage's 200-year-old Analytical Engine to life using the many plans he'd created of his design.

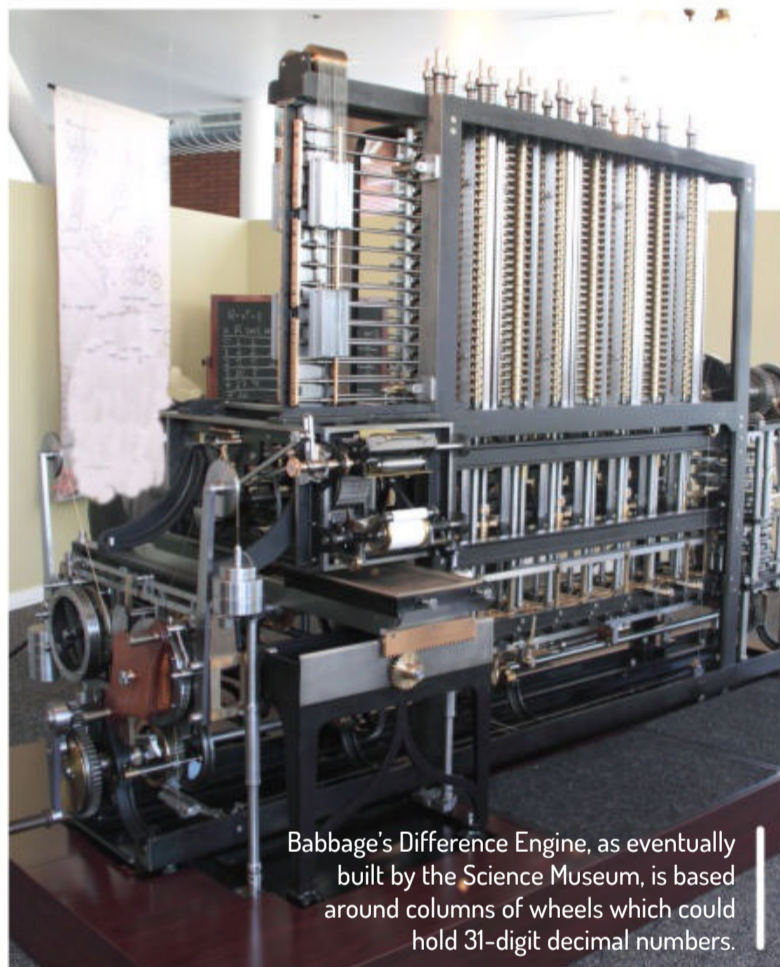


QUICK TIP

We've not even tried to describe the mechanical workings of the Analytical Engine, or even the Difference Engine. This is a case where a picture, or better still a video, is worth a thousand words. If you want to get a feel for this, take a look at the YouTube video <https://bit.ly/LXF283babbage> and the follow-on parts.

polynomial by Babbage's method of finite differences for values up to 2 in steps of 0.1, and plot them on the same axis as $\exp(x)$. The two traces will be almost indistinguishable until the last few points, and adding more terms to the polynomial will make it work for larger values.

Babbage's Difference Engine No. 2 – the version built by the London Science Museum – worked in much the same way as our spreadsheets, handling polynomials up to order 7 with a precision of 32 decimal digits. To do that it had eight columns of 31 wheels, each of which represented a decimal digit as defined by its rotational position. The column furthest to the right, the identity of which depended on the order of the polynomial, held the final constant difference, and the final result appeared at the left-hand column which was attached to a printer. Calculation of a new value required a lever to be cranked by hand by four revolutions, thereby carrying out the necessary four steps. Including the



CREDIT: Allan J Cronin, CC BY-SA 3.0, https://commons.wikimedia.org/wiki/File:Difference_engine.JPG

Babbage's Difference Engine, as eventually built by the Science Museum, is based around columns of wheels which could hold 31-digit decimal numbers.

printer, this remarkable machine contains about 8,000 parts, it weighs no less than 4.5 tonnes, and it's two metres tall by over three metres wide. Cranking it would have given the operator quite a workout.

The Analytical Engine

Despite never completing his Difference Engine, Babbage had bigger and better plans and here we must introduce the Analytical Engine. Again Babbage only ever built small parts of it, but his plans are comprehensive and paint a picture of a device that was far ahead of its time.

His terminology might not be what we're used to today, but many of Babbage's concepts were almost direct counterparts to what would be developed a century later. A key part of the Engine was its store and it doesn't take a great deal of imagination to realise that this was its memory. In particular, like the Differential Engine, it stored numbers in decimal notation by the rotational position of wheels, but this time as fifty 40-digit numbers. Then there was the mill, which in today's terms was the arithmetic unit, and which could perform the four basic arithmetic functions. Closely tied in with the mill was the Engine's barrels.

Using a principle borrowed from music boxes, these allowed functions to be defined internally by the presence or absence of pegs, and is pretty much the same as microcode in modern processors. Output was to a printer and Babbage also designed an associated curve-drawing apparatus, or pen graph plotter as we'd call it today. Input was from punch cards, like those used on the Jacquard weaving loom, and not dissimilar to those used on mainframes until the '70s, but threaded onto a tape to keep them in sequence.

Different sets of punch cards were used for different purposes, for example instructions and data. Having touched on instructions, the methods of coding are familiar today too. Included here, for example, are the concepts of conditional branching and, to use terms from Ada Lovelace's writings about the Engine, cycles and cycles of cycles or, in today's terminology, loops and nested loops.

Getting to grips with programming the Analytical Engine would take some considerable dedication and time. So we assume that this is really just for the real enthusiast today. However, trying out some programs on an emulator will give you some insight into the inner workings of the Analytical Engine and, should the bug bite, it'll be a good grounding for writing your own code.

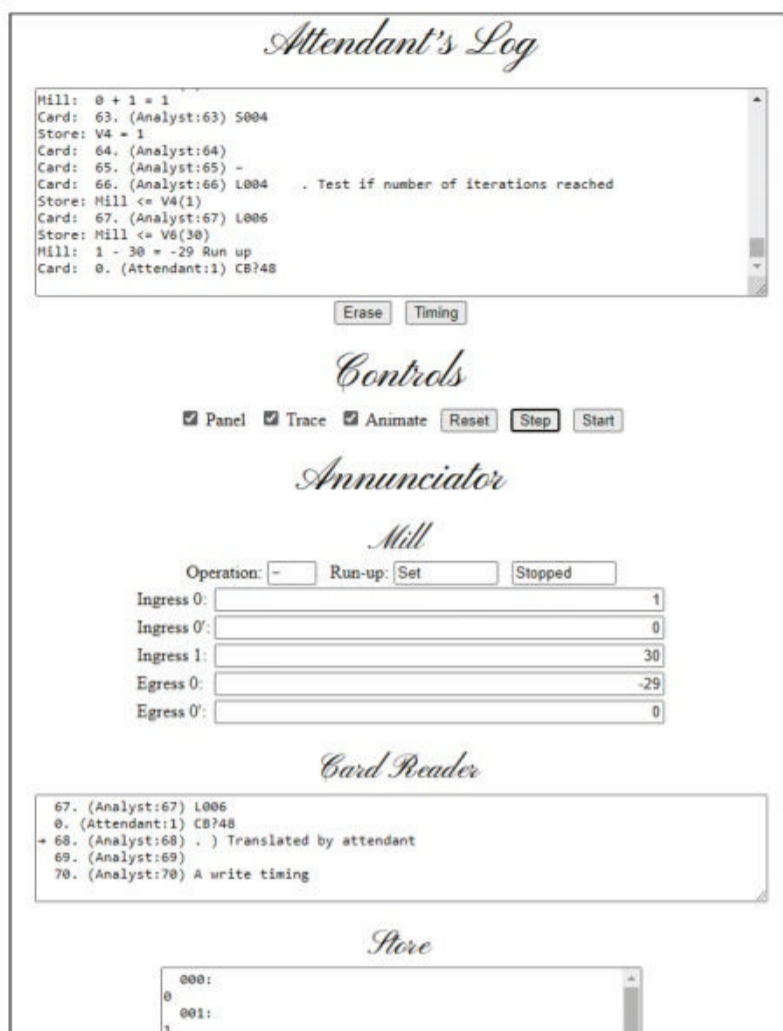
The emulator is written in Java and you can download it **from www.fourmilab.ch/babbage/cmdline.html**. However, at least to start, you might choose to use the Web-hosted version that can be found at **www.fourmilab.ch/babbage/aem** and execute one of the sample programs at **www.fourmilab.ch/babbage/sample.html**. Because the Fibonacci Sequence is well-known and easy to understand, we suggest you start with that program, so just click the associated gears icon and the code will be loaded into the emulator, ready to go.

Scroll down to the Controls area and click Start. The emulator doesn't attempt to provide real-time operation – and if it did you'd now have to wait a very long time – so the results will appear almost immediately in the Printer area. You'll easily be able to confirm that the

» THE FIRST PROGRAMMER

Following an introduction to Babbage, Ada Countess of Lovelace – who was the daughter of the poet Lord Byron – developed a passion that she shared with him for automated computing, and the Analytical Engine in particular. Famously, she translated into English the transcript by Italian engineer Luigi Menabrea, of a seminar that Babbage had presented at the University of Turin about his Analytical Engine. It was augmented by her own explanatory notes that were more extensive than the original transcript, and which provided an algorithm for the Analytical Engine to calculate Bernoulli numbers.

This is thought to have been the first published algorithm for a computing device and earned her the widespread recognition as the first ever computer programmer. It was sufficiently detailed that it was translated into C a few years ago and it worked, after one of Lovelace's instructions was corrected. So, it rather seems that as well as being the first programmer, she was also responsible for the first programming bug.



You can learn about the Analytical Engine using this Web-based emulator, or run it locally on your PC.

results are correct, but that exercise really won't have told you much about the operation of the Analytical Engine, so let's start again with the Fibonacci program newly loaded into the emulator. This time, before doing anything else, take a look at the Program area. This contains everything that's needed for the Engine to operate and appears to be at odds with our previous statement that different sets of punched cards were used for different purposes, such as data and instructions. The explanation is that the emulator differs from Babbage's original concept in this respect, to make it easier to use, while not adding any basic functionality that wasn't present in the original design.

The 'cards' in the Program area are presented in the order they'd be input to the machine. Cards that are prefixed with an N are number cards, which cause a particular value to be written to a specified location in the store. Variable cards are prefixed with L, Z or S, and load a value from the specified store location to the mill, load a value from the specified store location to the mill while zeroing the store column, and writing a value from the output of the mill to the specified store location, respectively. All other cards are operation cards and start with the character for the required operation, for example +, -, * or / for the arithmetic functions, < or > for left and right decimal shifts, and other characters for looping or flow control. Space doesn't allow us to describe them all, but they're fully described in the emulator documentation.

With this bit of background, you'll be slightly more informed about the operation of the Fibonacci Sequence program. But to see what's happening, ensure that Trace and Animate are ticked in the Controls

area and repeatedly click on Step instead of Start. Be sure to keep an eye on the Card Reader area so you can see which card is being read. You'll have to scroll up and down as you execute each step to see the changes in the relevant area – for example Printer, Mill or Store – although knowing what a card does will help you to know where you'd expect to see the result of that card's execution. There will be some operations that won't be obvious from our brief description above, but the comments in the code will help. And if all else fails, you'll have to take a look at the documentation at www.fourmilab.ch/babbage/cards.html.

Analytics yourself

When you've figured out how the Fibonacci Sequence program works – even if you've only managed to gain an overview – to see something of the sophistication of Babbage's creation, we recommend you take a look at the sample program entitled 'Plotting the Limaçon'. We strongly suggest you don't single-step through this because it'll take you a while – a very *long* while – but do take a look at the graphical output in the Curve Drawing Apparatus area. Even without single-stepping, it'll take some time for the emulator to complete its task, but that's not a patch on the 25 days it would have taken the actual Analytical Engine, as you can see if you take a look at the emulator's Attendant's Log area.

If you really think you ought to try writing your own code for the Analytical Engine, however simple – and we think you should – we have a suggestion for you. Recently, in our investigations into the 8080 processor, as used in the Altair 8800 personal computer and the 4004, the world's first microprocessor, we looked at programs to add together two number from memory, writing the result to another location in memory.

Simple as it might be, that would be a suitable first programming exercise and it would allow you to contrast the code with that which was used on developments that came more than a century later. **LXF**



It's a shame that the programming language that was named in her honour is no longer popular, but Ada Lovelace is widely recognised as the first computer programmer.

QUICK TIP

The Difference Engine's printer didn't only make a record on paper. It could also make imprints into a soft material that could be used as a mould for making printing plates. This allowed mathematical tables to be printed in volume, while eliminating the possibility of the typesetter introducing errors when transcribing data.

CREDIT: Public Domain, https://commons.wikimedia.org/wiki/File:Ada_Lovelace_portrait.jpg

» **TURN OUR COGS AND WHEELS** Subscribe now at <http://bit.ly/LinuxFormat>

Credit: www.neverware.com

Install Chrome OS on any PC hardware

Enjoy a simple-looking operating system with a lot of power, which can be added using a Linux container. **Matt Holder** is your guide.



OUR EXPERT

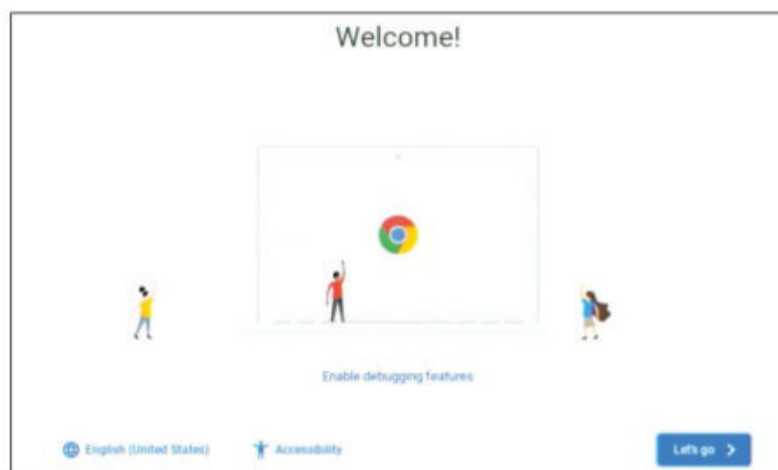
Matt Holder has been a fan of the open source methodology for over two decades and uses Linux and other tools where possible. In his spare time Matt enjoys listening to music and reading.

Chrome OS is based on the open source Chromium OS and over the years there have been various attempts to make Chrome OS builds available to any users who wish to install it on their personal devices. Around five years ago this author was attempting to use builds by 'ArnoldTheBat' on old eeePCs. While FydeOS is another Chromium OS build, which is targeted at the Chinese market.

This leads us nicely onto Neverware, which has been producing its own OS, called CloudReady, for around five years now. Neverware has a series of supported devices, which the OS has been tested on. Lots of other devices are compatible with the download as well, but haven't been tested.

Neverware's CloudReady has proven to be popular in the education environment because it can be installed on older (64-bit only) equipment and perform smoothly than when running better-known operating systems. Other benefits include being able to enrol CloudReady devices into the Google Workspace Admin console by purchasing a paid-for upgrade. With these enterprise-specific features, it's no surprise that Google acquired Neverware at the end of 2020.

With that background out of the way, it's time for us to discuss the Home edition, which can be downloaded and installed at no cost. At its heart Chrome OS is a heavily customised Linux distribution, with a very lightweight shell. In this article, we'll download and install the Home edition of CloudReady and then walk



This is CloudReady's welcome screen. This wizard simplifies initial connection to Wi-Fi, updates and displays T&Cs.

through the initial setup process, first login and familiarisation process.

Installing the OS

Neverware's CloudReady can be installed as a virtual machine or on bare metal. If running as a VM, only some of VMWare's products are supported. Running as a VM is outside the scope of this article, but further details can be found at <https://bit.ly/lxf283cloudvm>. Unfortunately Oracle's Virtualbox isn't supported due to incompatibilities with the graphics stack.

To install the OS on bare metal a USB stick needs to be prepared. This is a simple process and can be carried out using either the *USB Maker* tool, if running Windows or by following the steps below on Linux. By using the following link select the relevant option for your platform (Linux instructions will be discussed shortly): www.neverware.com/freedownload#home-edition-install.

Once the previous link has been visited, scroll to around half-way down the page and on the right-hand side, select the "Download 64-bit Image" button. While the image is downloading, scroll down the page and visit the link "Get Linux Instructions".

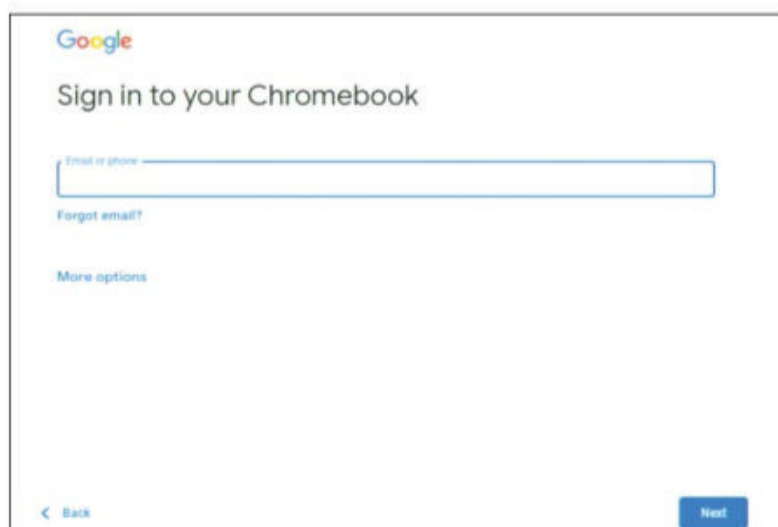
First we'll determine which device represents the memory stick plugged in. Open a terminal, plug in the memory stick (a minimum size of 8GB is required) and then run the following command:

```
dmesg
```

From this command there'll be a large amount of output, but should show references towards the bottom

QUICK TIP

Supported devices have been heavily tested by Neverware and can be seen at the link below. Other devices are likely to work as well, but won't have been tested and may have some hardware that doesn't work as expected. <https://guide.neverware.com/supported-devices>.



The standard login screen can be seen in this screenshot. A Google account is required to log into CloudReady.

of the output related to a memory stick, matching the size plugged in. This will also reference the partition if it's automatically mounted. For this article, an assumption is made that the memory stick is `/dev/sdd`, but this does need to be altered to fit your situation.

This can be double-checked by running the command `sudo fdisk -l`

The output from this should match the findings from the previous command. Once downloaded, extract the `.bin` file from the downloaded `.zip`. When the memory stick has been correctly determined and the file has been extracted, the following can be run. As a further note of caution, this command can wipe the operating system drive, so use it with care:

```
cd /PATH/TO/EXTRACTED/BIN/FILE
sudo dd if=NAME_OF_CLOUDREADY_FILE.bin of=/dev/
sdX bs=4M
```

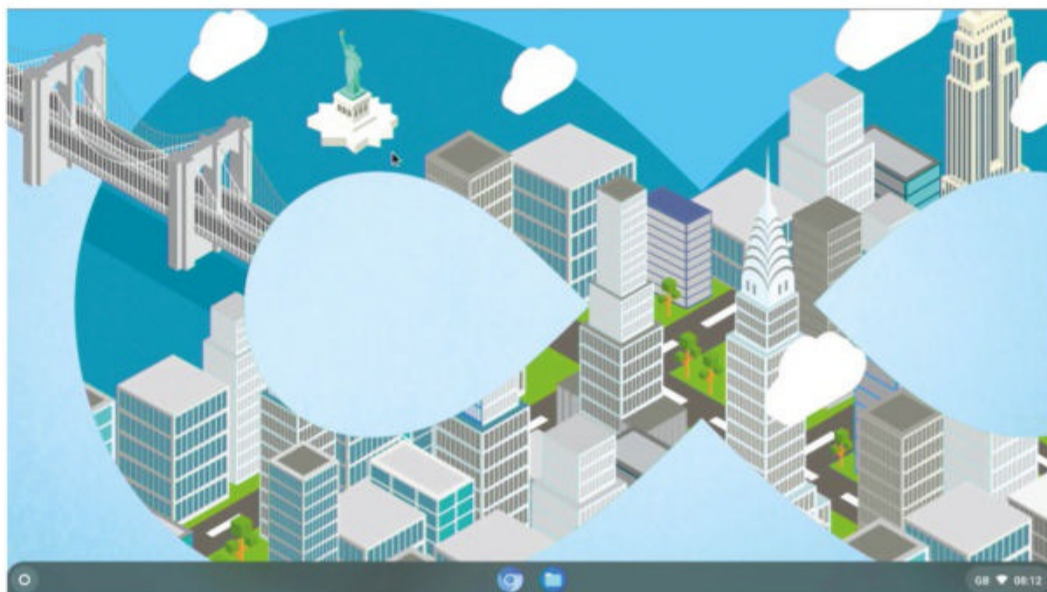
Once the file has been written, ensure that everything has been written to the memory stick with `sudo sync`

Installing CloudReady

Now that the memory stick has been prepared, the OS can be installed on your device. First, plug the memory stick into your computer and press whichever keyboard combination is required to enter the Settings screen. From experience, this could be F1, F2, Esc or Del. Some devices also have a physical button that can be pressed as the device turns on and this forces it to enter the Settings screen. From the Settings screen ensure that Secure Boot and Fast Boot are both switched off and that the boot mode is UEFI rather than BIOS. Once this has been done the boot order can be set to USB device first, or to boot from USB as a one-off override.

As the device boots, you will see a white screen with the logo and name of CloudReady in the middle of the screen. Depending on the speed of the device and USB stick being used, this may take a minute or two to boot.

Once booted, a plain login screen can be seen. Logging in doesn't need to take place yet because CloudReady can be installed from this point. Clicking in the bottom right corner will open a menu and one of the options is to Install OS. Select this option, read the warnings and agree to wipe the disk of the device being used. While being installed a progress bar will constantly whizz across the screen from left to right and a message will show stating that the installation should take no longer than 20 minutes. When the installation has completed, the device will automatically switch off.



Cloudready's desktop is pleasingly minimalistic and intuitive to use. From here the shelf, settings and launcher can be accessed.

Once installed, remove the USB stick (otherwise the PC or laptop may boot from there again) and switch the device on. If all has gone well then the device will boot with a white screen containing the CloudReady icon. A welcome wizard will then launch.

Follow the wizard to set the language and Wi-Fi details, and then you'll be asked to agree to the terms of service and asked to opt-in to providing diagnostic and usage data to Google. This last request is optional, so do what feels most comfortable. The final step of the process is for the device to search for updates. When this has completed a login screen will, again, appear. Congratulations – at the stage CloudReady is installed, connected to the network and ready to be logged into!

Log on with your Google account

Now, to use our new operating system. The first thing to do is to login with a Google account. Two-step verification options are supported on CloudReady and a phone will receive the push notification if this is configured. During first login follow the wizard and enable whichever syncing features are preferable. On first login a helpful welcome message will be displayed – spend some time learning about this latest release before closing the wizard.

Access the launcher from the circular icon in the bottom left-hand corner. This will open a screen with a search box (who'd have thought Google would provide a search facility?) where various programs can be searched for. The grey bar at the bottom of the screen is called the shelf and this is, in effect, the same as a dock

QUICK TIP

With experience of both Chromebooks and CloudReady in education, it was no surprise at the end of 2020 that Neverware was acquired by Google: <https://cloudreadykb.neverware.com/s/article/Neverware-is-now-part-of-Google-FAQ>.

» MAKE USE OF ACCESSIBILITY FEATURES

Accessibility features are built into CloudReady and are simple to access. By selecting the clock in the bottom-right corner of the screen a settings “cog” can then be selected.

Under Advanced and Accessibility lots of tools can be switched on or off. These tools cover areas such as offering a screen reader or select to speak, where the OS will read whatever's been selected and various display options, such as high

contrast mode, screen resolution changes and screen magnifiers. Other options available are keyboard options such as sticky keys and an on-screen keyboard. There are also options to make a mouse or touchpad easier to use if this would assist the user.

Another powerful tool to use is speech to text. This has been designed in such a way that dictation can be used in any field which supports text input. To switch

this on, the option can be found in the same place as the already described Accessibility features and is called “Enable dictation (speak to type)”. Once switched on, select where you want to enter text and select the icon of the microphone, which can be found in the Options area in the bottom right-hand corner. Keyboard shortcuts can also be used, but these will likely differ on the hardware being used.



in Linux distributions. Once applications have been opened they can be pinned to the shelf by right-clicking their icon and selecting the Pin option.

Clicking the bottom right corner will open the basic Options panel. From here the main Settings cog icon can be accessed as well as performing basic tasks such as switching Wi-Fi and Bluetooth on and off, pairing Bluetooth devices and casting the screen to any local Chromecast devices.

Much like Windows enabling seamless access to documents stored on OneDrive and Linux allowing access to files in certain cloud providers, CloudReady also gives you access to files in Google Drive. To open these files, open the *File Explorer* from the shelf, or search for *Files* in the launcher. You'll see that you can access files on the local device as well as files within Google Drive. If these files are native Google Docs then they'll open in the browser. So if, for example, these files are images then they'll open in the image viewer.

Applications are installed in a similar way to most OSs in recent years – from the launcher, search for Web Store. This will open a browser window and enable you to search for programs and install them. For example, an SSH application can be installed by searching for Secure Shell App. Once installed, this will then be accessed using the Launcher and can be pinned to the shelf. Some programs will open as separate windows and some will open as tabs in the browser.

Where's the Linux?

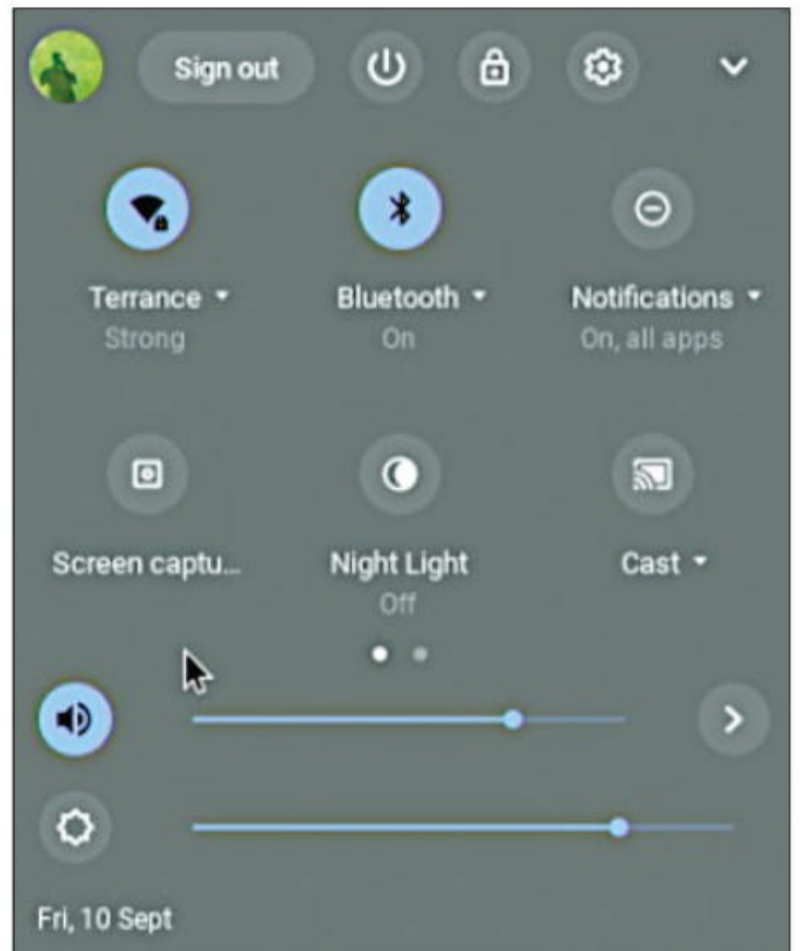
At its heart CloudReady is a Linux distribution with many of the same components in usage as other distributions. As we've already discovered the graphical environment is different from KDE, Gnome, Mate or any of the others we all know and use. A number of years ago work began to lay the foundations for Chromium OS (and therefore Chrome OS and CloudReady) to run containers and virtualisation. This project is called Crostini and further information can be found at <https://bit.ly/lxf283crostini>.

» CHROME OS VS CLOUDREADY

While Chrome OS is known for being a particularly lightweight operating system, in recent years some pretty advanced capabilities have been added. Within Chrome OS it's straightforward to install the Play store and then install any compatible Android apps. From experience this works fairly well, although not all GUIs designed for touchscreen devices work quite so well with a keyboard and mouse/touchpad. Unfortunately, this feature is not available within Neverware's CloudReady.

Chrome OS has the ability to link to an Android device via a Google account. This enables the sharing of notifications. Unfortunately, this is also not compatible with CloudReady. While not so well integrated, it is possible to use web-based versions of *WhatsApp* and *Telegram* to allow some notifications to be shared as well as messages to be sent from the device running CloudReady.

Neverware have also been known to deviate from the base Chromium OS project by offering experimental features, such as being able to use Oracle's Virtualbox and also install Flatpaks. These features have now been removed, due to the support being added for running a Linux distribution, which supports console-based and graphical applications.



CloudReady's basic settings can be accessed by selecting the clock, including Wi-Fi, Bluetooth, screen recording and volume tools.

The interesting part of this hard work is that, from within the CloudReady environment a Linux distribution can be installed, which gives you access to the device's file system and microphone, and for command-line and graphical tools to be installed and used. USB device support within the Linux environment is currently limited unfortunately, but can be worked around (to a point) by sharing a file location from CloudReady to the Linux environment. All that said, let's get our Linux environment installed, share a folder with it and install both a command line and graphical application.

Completing the installation is simple. First, select the clock in the bottom-right corner of the screen and then select the Settings cog. On the left-hand side of the window which opens, select Developer. Then, in the main part of the settings screen click the Turn On button, which is next to the title of "Linux development environment (Beta)".

When switched on a wizard will appear, which will carry out the setup. Follow through the wizard selecting a username and the amount of disk space, which will be allocated to the Linux environment. It's important to select the correct size during installation, but not critical. This size can't be directly changed, but it's possible to export from one Linux environment and import into another, which would enable the size to be changed. The wizard will then carry out the installation and a new terminal application will be installed. A terminal will open when the installation has been completed. The environment will be called penguin and the username will be the one that was chosen at setup.

What you're now accessing, via the terminal, is a Debian environment and this means that a large number of Debian packages can be installed. First, make sure the environment is up to date, which can be done by running the following commands:

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



Once installed the Linux container is accessed using the terminal. This controls the Debian container.

When completed, we can start installing applications. For this article, a command line and two GUI applications will be installed and these are called *NCDU*, *Remmina* and *Audacity*. *Remmina* enables connection to other devices using the RDP, SSH, VNC and SPICE protocols. *Audacity* is an audio recording and editing tool. Finally, *NCDU* is a tool that makes it possible to view disk space on a per directory basis. To install these run the following command:

```
sudo apt install remmina ncd u audacity
```

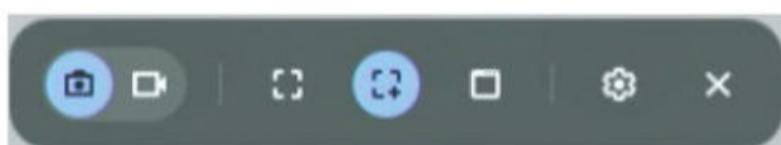
As previously mentioned it's possible to export settings from the Linux environment and this can be used either for back-up purposes, or to import into a new and larger environment. To carry this out, select the clock in the bottom-right corner and click the cog. Select the Developer section again and then open the Linux settings. From here you'll be able to export settings using the Backup & Restore section and then they can be saved to the device. If completing an import into a larger environment, follow the installation steps again and then use the Backup & Restore options from the Linux Settings screen.

While the Linux Settings screen is open, it's a good time to explore the other options. From this screen shared folders can be viewed and deleted. To share the folder in the first place, one must first open the File Explorer, navigate to the correct location, right-click and select the Share with Linux option. Some USB devices can also be shared, but this is limited to certain devices.

It's good to share

To share a device, plug it in, open the USB sharing options and if available select the toggle to share with Linux. The Port forwarding option can be used to allow ports to be forwarded between the physical device's IP address and the Linux environment. In this way a web server could be set up, for example, and web content could be shared for testing purposes. The final option to mention is that of microphone support. When selecting this option the microphone will then be allowed within the Linux environment. This could be useful if running *Audacity* for audio recording and editing purposes, say.

When logging into CloudReady, Linux can be launched by opening the Launcher and searching for Linux or Terminal. This can also be pinned to the shelf. To open command line applications, the name can be entered within the Terminal application. To view disk utilisation with *NCDU*, simply type `ncdu /` and press Enter. This will then display a simple bar graph of folder



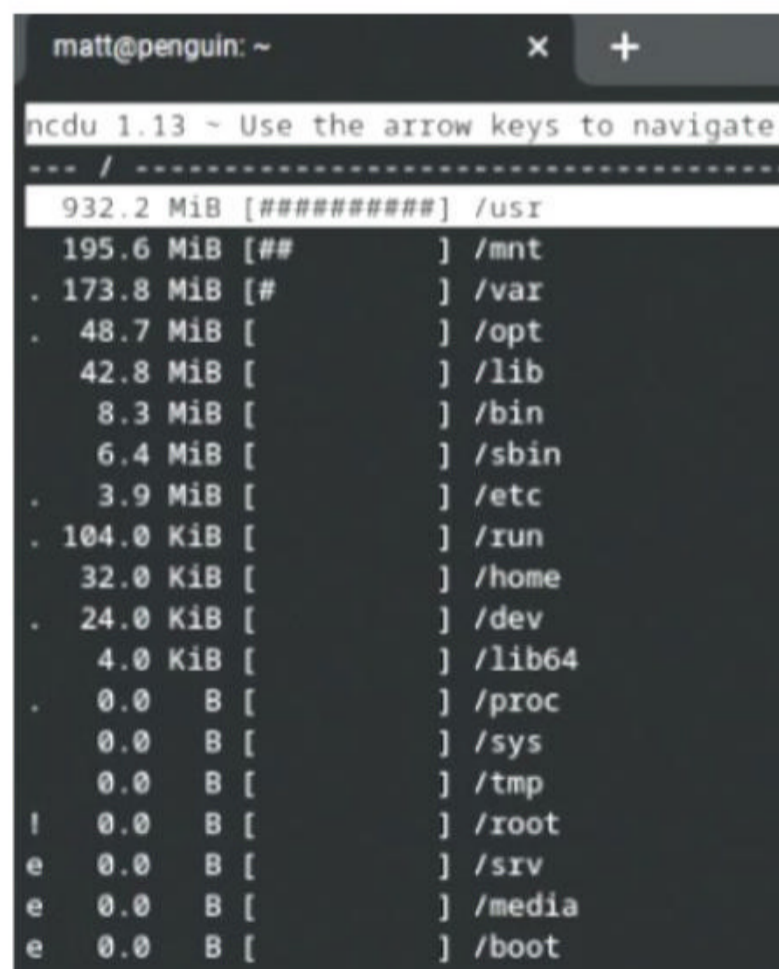
CloudReady includes a simple-to-use screen recording, while screen capture is built into CloudReady.

utilisation by file size. Note that this will display the utilisation of files within the container, rather than the disk usage of CloudReady itself.

Graphical applications are well integrated into the system and will appear in the launcher. To access them, open the launcher and search for the name. Once *Remmina* has been opened its usage is identical to that on other Linux distributions and connections can be added to devices via RDP, VNC or other protocols. The final application to use is *Audacity*, which was installed earlier on in the article. Open this in the same way as *Remmina* and don't forget to enable access to the microphone if using it for recording audio. By selecting the Record button, the default settings should be correct and will allow audio to be recorded. When saving files from applications within the Linux ecosystem, these can be accessed from the *File Explorer* under the Linux files section on the left-hand side.

For the more adventurous out there the Debian base environment can be swapped out for Arch. While it looks like both containers can exist side by side, it's probably safer to assume that, if completing these instructions, the Debian environment will be removed and any installed applications will be lost. The instructions and explanations of each command are taken from the excellent Arch wiki and can be found at https://wiki.archlinux.org/title/Chrome_OS_devices/Crostini.

The aim of this article has been to introduce a lightweight and very usable operating system, and explain how some of the more advanced features can be used. With that said – enjoy! **LXF**



Here's the ncurses application, NCDU, running within the Linux container that's displaying disk usage.

QUICK TIP

The Neverware CloudReady knowledgebase offers a wealth of information about the OS, covering topics from running as a VMWare VM to how the release numbering works. See <https://cloudreadykb.neverware.com/s/>.

» **INSTALL US THE EASY WAY...** Subscribe now at <http://bit.ly/LinuxFormat>

Run a Ghost blog on your server

David Rutland would like to apologise in advance for the supernatural puns contained herein – but it was written in the run-up to Halloween.



OUR EXPERT

David Rutland is a tinkerer and a dilettante. He buys domains on a whim and runs them from a Raspberry Pi behind the couch..

The internet is a bleak place in the third decade of the 21st century. Away from the walled-garden shouting matches and ego-stroking of the social media sites, you'd be forgiven for thinking that the bulk of the web is an endless desert of SEO optimised buying guides, VPN affiliate sites and filth.

Well, that's our experience of it anyway. Gone are the days of stumbling around from site to site, following links from personal blogs to whatever whimsical sites the authors care to link to. But it doesn't need to be that way. Break away from the usual routine and well-trodden paths, and you'll find that there are still a few pockets of genuine authentic content, written by authors who want to document their lives without plastering it across Facebook for a handful of Likes, or document their hobbies and passions without selling out to a huge media conglomerate (*hey, that's us!—ED*).

Yes folks, independent blogging is back with a vengeance, and with your shiny new VPS, you can be a part of it too! Imagine the delight of a total stranger as they stumble across your blog and are sucked down into your thoughts, musings and oh-so-witty take on life, politics, technology and pets.

Picture their glee as, having bookmarked your words of wisdom and shared it with their friends, they move on to yet more discoveries through your expertly curated set of links to other blogs worth their time.

So what is a blog? You know damn well what a blog is. The correct question is, "How do I choose what blogging software to use?" with the possible additional question, "How do I set it up?" We'll address both now.

Down at the most basic level, your blog can be a simple text document written in *nano*, *mousepad*, *kate* or whatever. You can put it in the root of your VPS and update it by simply adding more text to the top. It works, it's simple, it's super-quick and it's an almost no-knowledge of way of getting your thoughts out into the world. The downside is that a text file is limited to text only. No images, no links, and your blog is unlikely to be discovered by accident or happenstance.

The next possibility is to do it in HTML by hand, with the added option of using a Static Site Generator (see *Roundup LXF282*) such as Jekyll to make creation and updates a walk in the park. But SSGs are last year's hot topic, and while they hold a treasured place in the hearts of everyone at *Linux Format*, it seems like a severe underuse of the resources of our mighty VPS.

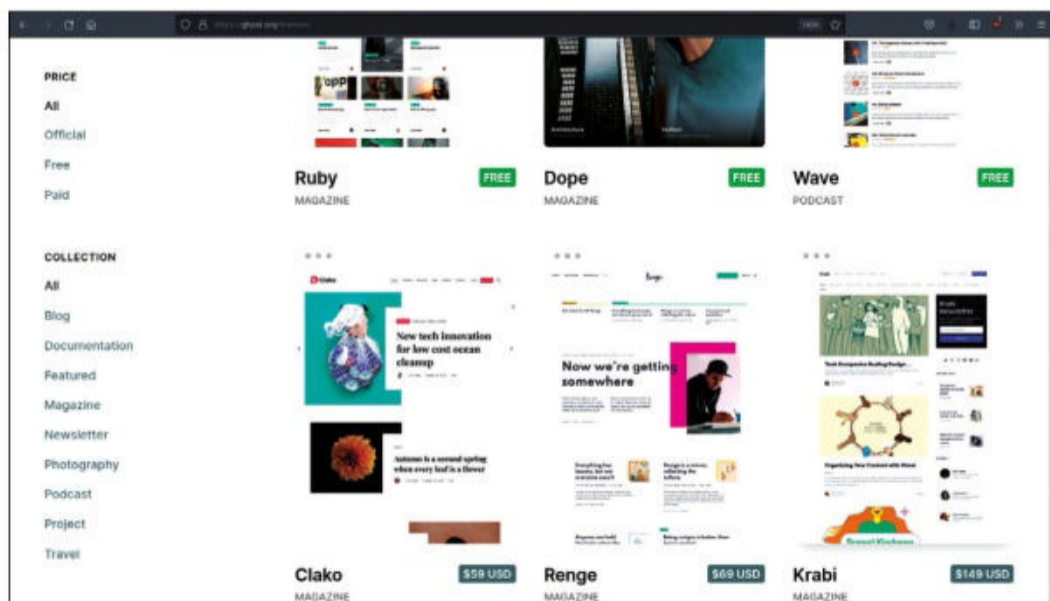
WordPress, schmerdPress

What we want is a content management system. In contrast to static sites which require a certain amount of HTML or markdown knowledge and a hefty dollop of organisational skill, a CMS takes care of all almost all of the nuts and bolts which result in what you see on the resulting webpages.

It's entirely possible to create and manage a website without any knowledge of anything beyond the words you want to write about your chihuahua's pregnancy, and the adorable (or yucky) pictures you choose to illustrate the same.

Beyond a few hold-out sites which are stuck in the Stone Age (*linuxformat.com?—ED*), almost every website in existence uses a CMS. The one you'll most likely have seen most of – or even have experience with yourself – is WordPress. It's simple, it's free, and installation is as easy as *wget*-ting a file into your document root and unzipping. It doesn't get much simpler than that, which is why it is the power behind more than one third of the sites currently in existence.

But in the same way that viruses are usually tailored for Windows systems, the very ubiquity of WordPress makes it vulnerable to attack. If you've ever run a



The Ghost Marketplace is where you go if you want to get new themes to change the look of your blog. We found it to be an excellent opportunity for procrastination.

website, whether WordPress or not, you will have noticed from your access logs that most of the automated login attempts are tailored for WordPress, and if you really feel like descending into a nightmare spiral of paranoia, search “WordPress hack” on YouTube and watch as ‘security experts’ breach the admin dashboard and manage to bring up a shell with root access to the host system. Then there are the problems with plug-ins, where villains will buy up a company behind a popular one and turn users’ WordPress blogs into malware-spewing trap sites. In short, WordPress attracts hackers like our late, much-lamented cat attracted fleas.

Woooooo, a Ghost!

Ghost is a Content Management System every bit as competent as WordPress, and extra spooky. In our expert opinion, Ghost’s low penetration into the Open Source CMS market makes it a much lower profile target for hacks. It also has some killer features that WordPress is lacking.

Vulnerabilities aside, Ghost is (in our opinion) a far better blogging platform than WordPress, because that is its sole raison d’être. You won’t find any e-commerce plug-ins for Ghost, and everything about the interface is dedicated to creating a comfortable writing experience for the blogger, and a clean, well laid-out reading experience for your audience.

Who ya gonna call?

Have we managed to convince you yet? Great! You now know what Ghost is and you’ve decided it is the medium through which the world needs to hear your words.

We’re going to assume that you’ve read and thoroughly digested our previous two VPS features in **LXF281/ LXF282** (<https://bit.ly/lxf281lxfserver> and <https://bit.ly/lxf282lxfserver>). If not, don’t worry, read them at the Archive and then get back to us. All done?

Next you need to decide on a domain or subdomain for your ectoplasmic blog. We went with **ghost.lxf.by**, and followed our VPS provider’s instructions to link it with the VPS. Unlike WordPress, Ghost does not simply allow you to dump a bunch of files into the document



While we’re loathe to criticise anyone who worked on the project or their design choices, it would be remiss not to point out that the default colour scheme is a crime against humanity.

root, point Apache in its general direction and access it through ports 80 and 443.

As with an increasing amount of web-facing software, Ghost takes a more ethereal approach, operating on its own port – in this case 2268. Your server will need to proxy incoming traffic on 80 and 443 so that it finds its way to 2268.

If you failed to follow our advice in the first tutorial of this instalment and chose Nginx over our heartfelt recommendation of Apache, then congratulations! The Ghost installer script is designed to configure an installation on Nginx automatically. Feel free to skip ahead. Apache users will need to activate a couple of mods to enable Apache to do the proxying:

```
sudo a2enmod proxy proxy_http
```

and then restart Apache:

```
sudo service apache2 restart
```

You’ll then need to create a **conf** file so that Apache knows what it’s supposed to be doing.

```
sudo nano /etc/apache2/sites-available/ghost.conf
```

Inside the **conf**, you will need to tell Apache what port to listen on, the name of the server, and what port to proxy requests to. Ours looks like this:

```
<VirtualHost *:80>
ServerName blog.lxf.by
ProxyPass / http://127.0.0.1:2368/
ProxyPassReverse / http://127.0.0.1:2368/
ProxyPreserveHost On
</VirtualHost>
```

Then enable the **conf** with `sudo a2ensite /etc/apache2/sites-available/ghost.conf` and restart Apache once more. You will need to give Ghost its own database,

QUICK TIP

You may find (as we did) that some of your standard terminal shortcuts may not work with your VPS. Ctrl+L clears the screen on our home machine, but results in a mess when connected to the VPS. We had to go old-school and type out the word ‘clear’.

» OTHER DEDICATED BLOGGING SOFTWARE TO CONSIDER

WordPress and Ghost aren’t the only Content Management Systems available for your Virtual Private Server, but we really like how powerful and easy to customise they are, as well as the support available from their online communities. The web is a varied place and no one size fits all. If Ghost doesn’t tickle your fancy and you’re turned off by the ubiquity of WordPress, there are hundreds of alternatives for you to try.

Our favourites from among the rest include BigTree CMS (www.bigtreecms.org), a straightforward, well documented and capable CMS written with PHP and MySQL. Silver Stripe (www.silverstripe.org) advertises itself as “the intuitive content management system and flexible framework loved by editors and devs alike”. REDAXO (www.redaxo.org) comes with some fabulous features, but the documentation is written entirely in

German, so unless your fairly fluent, you should probably stay away. October CMS (<https://octobercms.com>) is based on Laravel to offer an excellent modular system and plenty of free plugins.

New projects are coming online all the time as users wanting a very particular feature set create their own forks or create their own CMS from scratch, so it’s always a good idea to keep an eye on the news.

QUICK TIP

Some VPS providers such as Digital Ocean offer a one-click web install for Ghost, meaning that you can skip practically all of these steps. You could take the easy option, but where's the fun in that?

so open up Maria with:

```
sudo mariadb
```

Then enter each of these commands which will create a user called `ghost` on a new database called `ghost` and allow the new user to use the new database. Feel free to choose your own wacky names.

```
CREATE DATABASE ghost;
CREATE USER ghost IDENTIFIED BY 'secretpassword';
GRANT USAGE ON *.* TO ghost@localhost
IDENTIFIED BY 'secretpassword';
GRANT ALL privileges ON ghost.* TO ghost@
localhost;
FLUSH PRIVILEGES;
quit;
```

Remember to keep the details safe and secure for later, and not on Post-it note stuck to your monitor.

To actually install the Ghost CMS software, there are only two additional dependencies: `node.js` and `node package manager (npm)`.

Unfortunately the versions of `node` in the default repositories are not compatible with Ghost, nor is the latest version (v16.10.0). You will need to install the `nodesource` PPA first. Head over to your home directory and do:

```
curl -sL https://deb.nodesource.com/setup_14.x | sudo
bash -
```

Run `sudo apt-get install -y nodejs` to install Node.js 14.x and `npm`. Note, there is no need to run `apt update` before you do this as it is run as part of the setup script. Check that the install was a success by querying the version number with:

```
node -v
```

and your VPS should return a response of:

```
v14.18.0
```

Now it's time to install the tool which will install Ghost:

```
sudo npm install ghost-cli@latest -g
```

After a few seconds with a swanky ASCII progress bar, you'll be back on the command line. Finally, we can actually run the command which will (we promise) install Ghost.

```
ghost install
```

In the time it would take to hold a reasonably short seance (or a cup of tea) with the spirit of a deceased pet, the install will suck down everything it needs and at long last get installation underway.

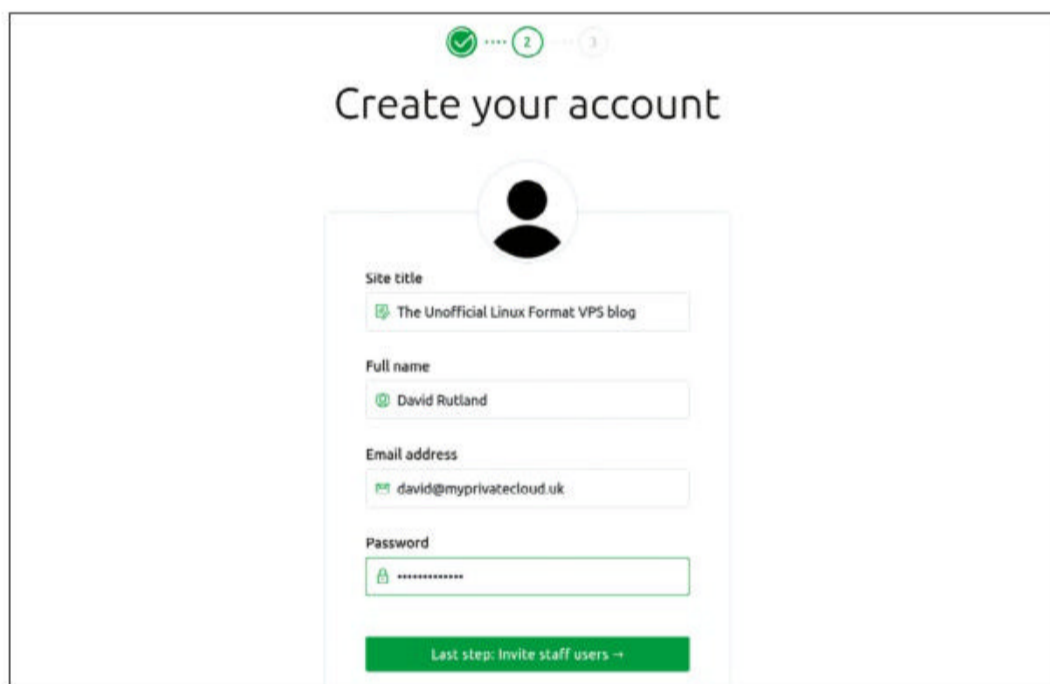
Nearly there...

You'll be asked for your blog URL (enter `http` rather than `https` as we have not yet set up our SSL certs), your MySQL hostname (`localhost`), your MySQL username and password.

The more keen-eyed among our readership will have noted a number of warning messages and skipped steps during the setup. These (mostly) relate to Nginx, which you will recall is not present on the **LXF** server as we opted for Apache instead. If you did choose Nginx, all should be peachy.

Visit your new blog at the domain you specified and fill in the name and email address fields and think up a cool title. Then close the tab and head back into terminal again because it's time to secure your blog against the ne'er-do-wells who want to listen in on your traffic and attack your readers.

With security topmost in our mind, we ran `sudo certbot`, entered our email address and declined to join the EFF mailing list (because we're already on it). Certbot gave us a list a short list of domains for which certificates could be obtained.



This is the first interaction you'll have with the Ghost web interface. It's super-pretty, and straightforward too.

» THE GHOST WITH 1,000 FACES

As with any new toy or gadget, it's easy to become bored quickly once you've mastered the basics, and you want more things to tinker with. Understandable; it's either that or crack on with the hard work of writing 3,000 word posts on the nature of your own personal brand of spirituality.

Fear not! You can put off the inevitability of doing any actual writing work by giving your blog a complete makeover. Ghost's overall look is governed by themes. The default theme is called Casper (for obvious reasons), and to our mind, it's pretty good. Everything is laid out nicely, and it's

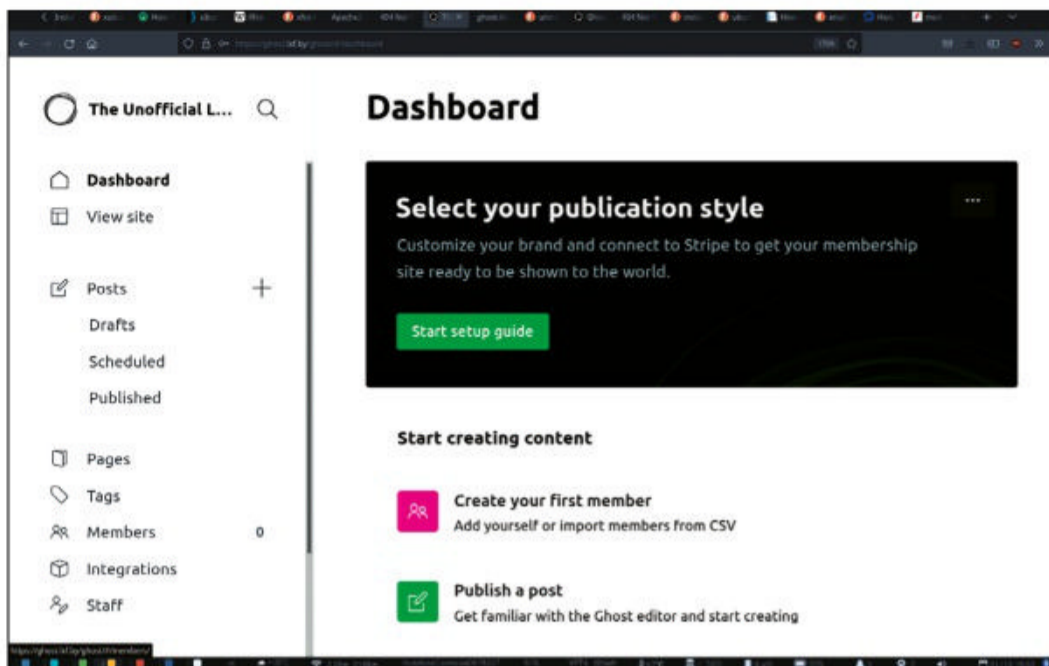
a pleasure for your visitors to view. Head over to <https://ghost.org/themes> and you'll see hundreds of themes which can completely change the way your blog appears. There are themes suitable for photo galleries, bulletins, newsletters, travel journals and more.

Some of these themes are free. Most are not, and the highest price we saw on the marketplace was (gulp) \$149. We're not criticising here – people work hard to create these themes, and deserve to be compensated for their toil – but we chose London from the 24 available free themes. We're not suggesting that we're

cheap here at **LXF** Towers, but, well, we drink the free coffee offered here.

The creators of the London theme describe it as "A bold, minimal theme for Ghost, focused on clean typography & beautiful imagery", and it is. We installed it by the simple expedient of clicking install on the theme's page, entering the address of our blog, clicking install again, then activate. Honestly, it couldn't have been simpler.

In our opinion, the new theme looks better than fabulous. Now all we need to do is actually write some fascinating copy for our legion of fans.



The Ghost dashboard makes it super-easy to customise the look and feel of your blog. Just hit the big green button to get started.

We selected the relevant one, and opted for automatic redirection – meaning that anyone typing blog.lxf.by or http://blog.lxf.by would be magically transported over to <https://blog.lxf.by>. That's it. Done. Simple.

Ghosts in the haunted mansion

Cold shivers down your spine, and mysterious creaking doors aside, Ghost is a pleasure to use and also incredibly customisable.

The default page comes pre-populated with a headline story, with the oh so evocative title of “Start here for a quick overview of everything you need to know”, and six equally spooky stories including such thrillers as “Customizing your brand and design settings”, the spin-chilling “Building your audience with subscriber signups” and the terrifying “Selling premium memberships with recurring revenue”.

We're sure that some of our readers would love to avail themselves of these capitalist pieces, but it's not us. We're just here to build a cool-looking blog to show off our burgeoning bunch of bonsai bushes to the world. The layout is nice though, and the articles on customisation are certainly worth a read.

On the dashboard immediately after you log in through your.site/ghost, there's a large black banner

with a green button marked 'Start setup guide'. This is the best first step you can take to alter the look and feel of your Ghost blog, and clicking it will take you on a step-by-step journey through the aesthetic details. We chose to do away with the fuchsia accent highlight in favour of a dark, mysterious black, and the background pastel gradient image was tossed away in favour of a *Matrix*-esque image of green zeros and ones cascading down a black background. Very Noughties.

The front page was still looking dull with the default non-images which came bundled with the installation, so we downloaded a fistful of Linux-related images with Creative Commons licences, and

dived into the post editor to switch them over.

Get writing

Writing in Ghost is simple and intuitive. Creating a new post is as simple as pressing the + symbol on the left-hand side. The layout is a plain white page, with an area for the title and another area for the text.

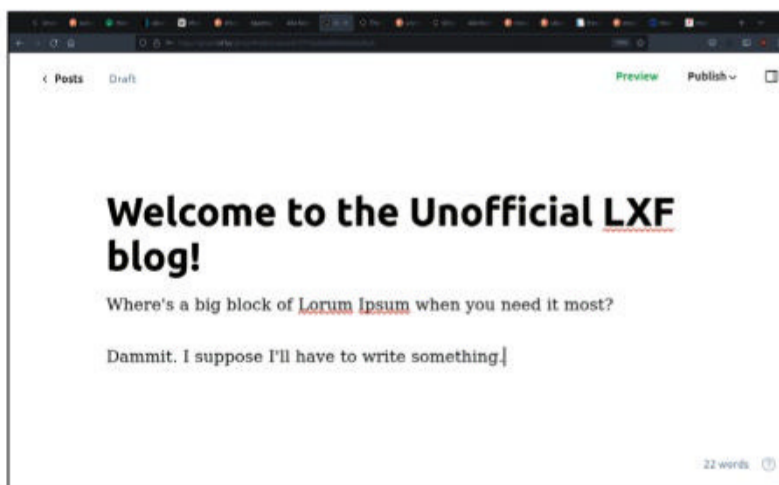
Additional elements can be added simply by pressing the + button prompt and choosing whether you'd like an image, a text block, HTML, a gallery, or embeds for the most common media providers: YouTube, Spotify and so on. If you've ever used the premium 'WP bakery' WordPress plug-in, you'll be instantly at home – except moreso, because Ghost is considerably quicker and a lot easier to use.

Over in the Integrations section, you'll find that Ghost plays nicely with a huge range of tools, from Plausible analytics to Slack. The amount of difficulty involved in setting up each of these varies, but does tend to be on the very simple side.

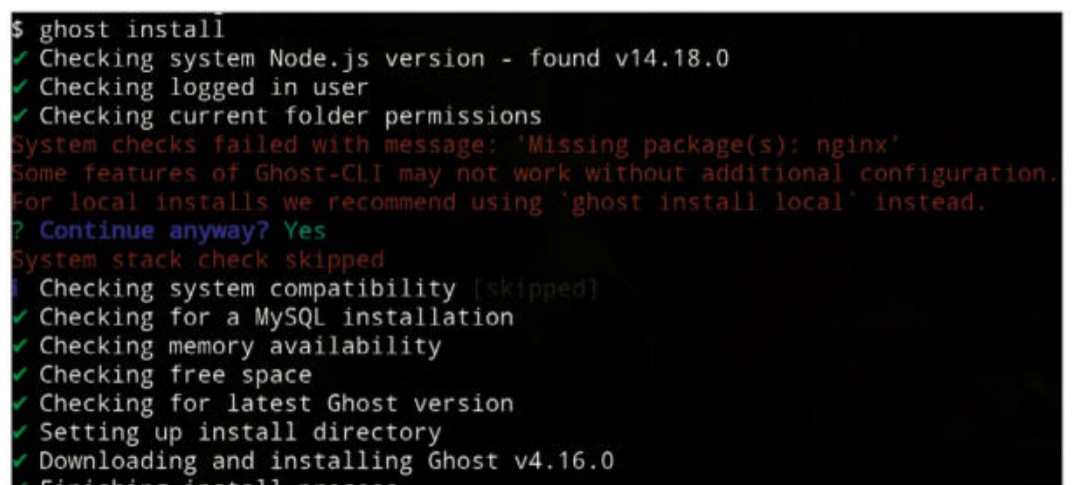
Hopefully, by now you'll have a good-ish feel for the way Ghost works, and can find your way around it without too much difficulty. For such a sophisticated piece of software it's surprisingly easy to get a grasp on how it works – which is the true benchmark of a great project, we suppose. **LXF**

QUICK TIP

When publishing to the web, you need to consider copyright concerns. You should only use images under a Creative Commons licence (and abide by the terms of that licence), or which are in the public domain. We're not joking – some photographers make more by suing over copyright issues than by selling photos. Try Flickr or Wikimedia Commons.



Writing in Ghost is simple and clean. We'd say it was a joy if we could think of something interesting to write about.



Once the prerequisites are out of the way, installing Ghost is a one-liner followed by a filling in a few text fields. It's even easier if you're using Nginx instead of Apache. Sorry about that.

» GET EVEN MORE SPIRITUAL AND Subscribe now at <http://bit.ly/LinuxFormat>



The home of technology

techradar.com

HotPicks



Alexander Tolstoy

has some crazy video conferencing ideas after setting up his fake webcam background...

Glyphtracer » OnlyOffice » Open Video Downloader » gThumb » JPEG XL » Annotator » Monitor » SpaceCadet » A/B Street » Metadata Cleaner » Fake Background

FONT EDITOR

Glyphtracer

Version: GIT

Web: <https://github.com/jpakkane/glyphtracer>

Most people don't ever bother making their own fonts because of the many complexities that are usually involved in the process. Not only do you need to be able to draw professional-looking letters, but you also need to craft spaces between them, think about every last little detail and finally assemble the whole lot into a font file using dedicated font-editing tool. However, thanks to *Glyphtracer* there's now a shorter path and a genuine opportunity to create your very own font in minutes. That is perhaps why this tiny piece of software is one of the most inspiring *Hotpicks* we've ever discovered.

Glyphtracer is a Python application that assists you with creating your own font based on a bitmap image. You can simply draw characters in a painting application, feed them to *Glyphtracer*, assign letters to each drawn character, then export the results as an SFD file. You can open the resulting file in *FontForge*, optionally edit it there and export it to TTF or another font format via File>Generate Fonts.

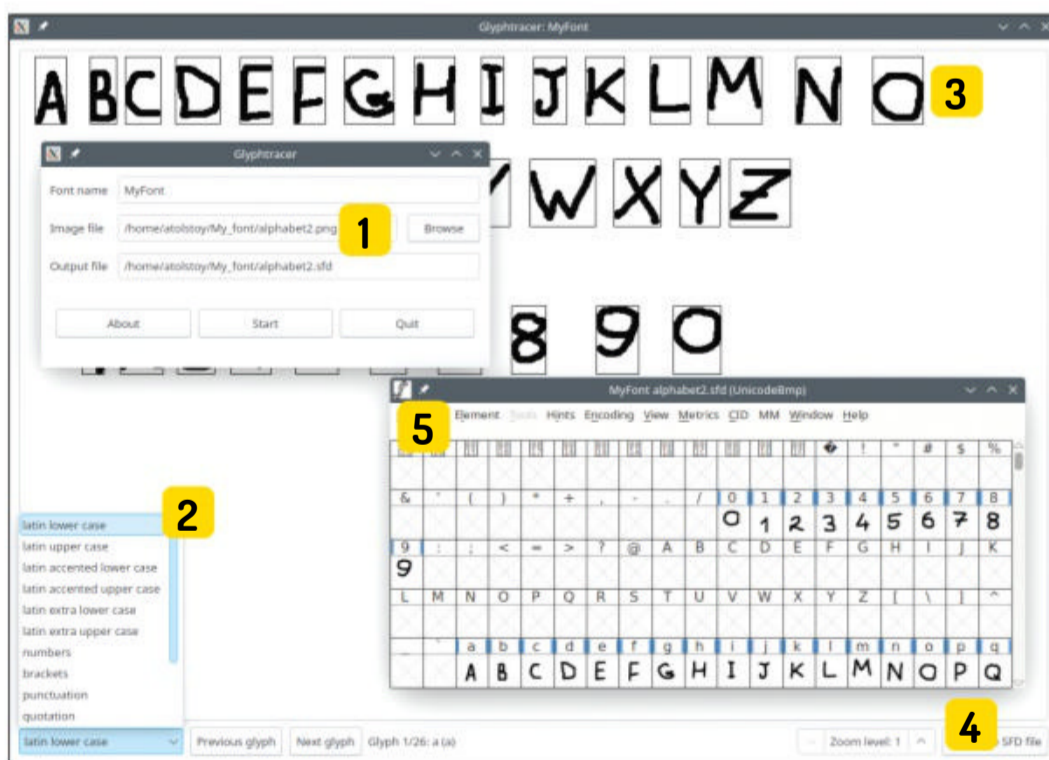
How does *Glyphtracer* know which character is which? Well, it doesn't, so it asks you to do the job. The lower part of the tool's window contains the current set of characters to be assigned. By default, it's in Latin lower case, which is exactly what we need to get started with our deliberately simple, amateur hand-drawn font. So, if your drawing contains letters in alphabetical order, all you have to do is sequentially click one box after another to make *Glyphtracer* assign the correct letters.

It's also possible to populate a font with upper-case letters, brackets, punctuation marks, numbers and other sets of characters to make it more complete. Of course, *Glyphtracer* is not a commercial-grade font tool and it certainly won't suffice for professional artists who spend months or even years crafting Antiqua, serif or Art Deco-style fonts. If your goal, however, is simply to convert a drawing into a TrueType font for your own entertainment or educational purposes, *Glyphtracer* is a great find. Give it a try.



Creating your own font from scratch has never been so easy and straightforward, thanks to *Glyphtracer*.

EXPLORING THE GLYPHTRACER INTERFACE



- 1 Prepare a bitmap drawing**
This should be an image file (PNG, BMP and so on) with indexed colour mode and just one single colour (without gradients).
- 2 Select a character set**
Here you can switch the assign tool mode to different sets of characters, such as upper case, numbers, brackets and so on.
- 3 Assign letters to these boxes**
Look at the bottom bar to see the current letter and click the appropriate box. Repeat for the rest of the boxes.
- 4 Save results to an SFD file**
Glyphtracer produces a semi-complete font. You need to finalise it in a font editor, such as *FontForge*.
- 5 Create a real TrueType font!**
Use the *FontForge* File menu to create a TTF file. Then install it and give it a trial run in a text editor, such as *LO Writer*.

OFFICE SUITE

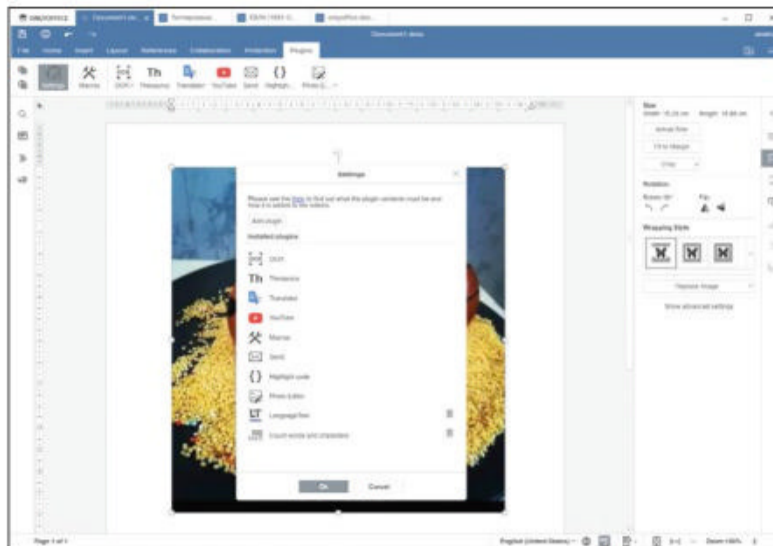
OnlyOffice Desktop

Version: 6.4.1 Web: <https://github.com/onlyoffice/DesktopEditors>

You should already be accustomed to the *OnlyOffice* bundle thanks to the plethora of reviews and tutorials *Linux Format* dedicates to this outstanding office suite. It seems to hold a deserved second place after *LibreOffice* in terms of capabilities, performance and ease of use, which were all thoroughly tested in the office suite *Roundup* in **LXF279**.

The team behind this software maintains a snappy development speed and frequently releases new versions, so now we have *OnlyOffice Desktop 6.4.1*. Another set of improvements landed this time, such as auto-capitalisation of the first letter in a sentence, a new document review mode, the new sparkline object in *Spreadsheet*, support for importing TXT and CSV files, and more. If you already use *OnlyOffice Desktop*, be sure to update to this version to enjoy all the additions.

The suite may not be perfect – it tends to load and save files slowly, and the UI doesn't always feel smooth



Don't forget you can extend the office suite with extra plugins and get great new features.

– but it's a well-developed, mature office suite that handles even complex *Microsoft Office* files like a charm, carefully retaining every bit of formatting.

OnlyOffice also has a breathtaking set of default plugins, including an image editor, YouTube integration, a translation service and even a baked-in OCR module. The web version of *OnlyOffice* ships with even more plugins, but the desktop edition's standard set is already very generous, and missing plugins can always be installed manually. Locate and grab a plugin from *OnlyOffice's* GitHub page, repack its files as a ZIP archive, then rename it so it has the .plugin extension. After that, simply go to Plugins>Settings and locate your plugin file. Hopefully, some day the *OnlyOffice* team will roll out an automatic plugin store integration to save us from doing it manually.

VIDEO DOWNLOAD TOOL

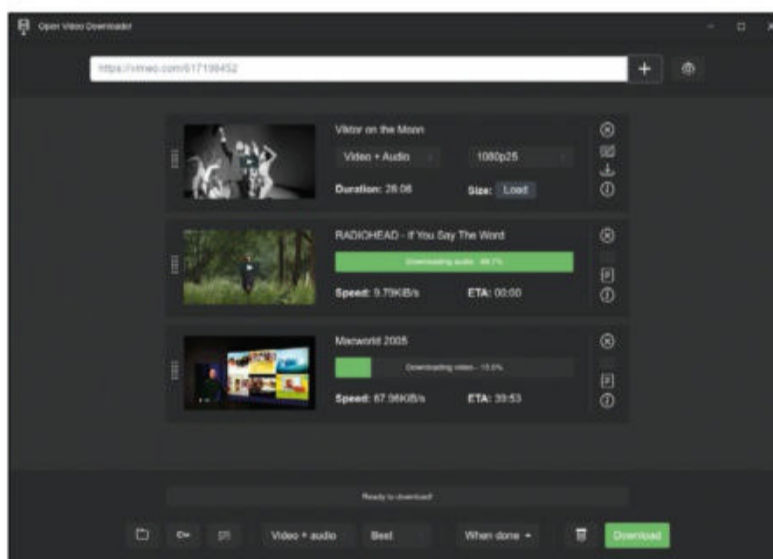
Open Video Downloader

Version: 2.3.1 Web: <https://github.com/jely2002/youtube-dl-gui>

Once it looked like the quest for the perfect front-end to the extremely useful *youtube-dl* command would last forever. Possibly because video-hosting services usually have far too many unskippable ads that nobody wants to see. We're pleased to say we've found a new, fanciful GUI that can serve all your video download needs.

Open Video Downloader does its job with style and no distractions. In the endless stream of *youtube-dl* front-ends, it looks more polished and takes little effort to get running. We downloaded the AppImage bundle and ran it with no issues. The program features a fancy dark-themed GUI and feels like a typical modern tool built with *Electron* and *Node.js*.

Visit any video-hosting service that *youtube-dl* supports (that's almost everything, not just YouTube) and copy the video link to the clipboard. *Open Video Downloader* fills in the link input field for you, so just



Download more videos from the internet with ease. Just put them all in the queue.

press the plus button. The video isn't downloaded instantly, enabling you to add more entries to the queue. You can then rearrange entries and review video details. Then either hit the download icon next to the video, or use the green button to launch the whole queue.

We liked the number of extra settings you can change within the application to get the right download type. For instance, each entry in the queue enables you to quickly choose if you want video and audio, or just one part of the pair, what quality it should be, or if you need subtitles (if available). Once a download is complete, hit the Show In Folder button to open a file manager (by default, all files land in **~/Downloads**), or the Open File button to go directly to the video. *Open Video Downloader* is a feature-rich, intuitive tool that even a non-tech-savvy person can quickly master.

IMAGE VIEWER

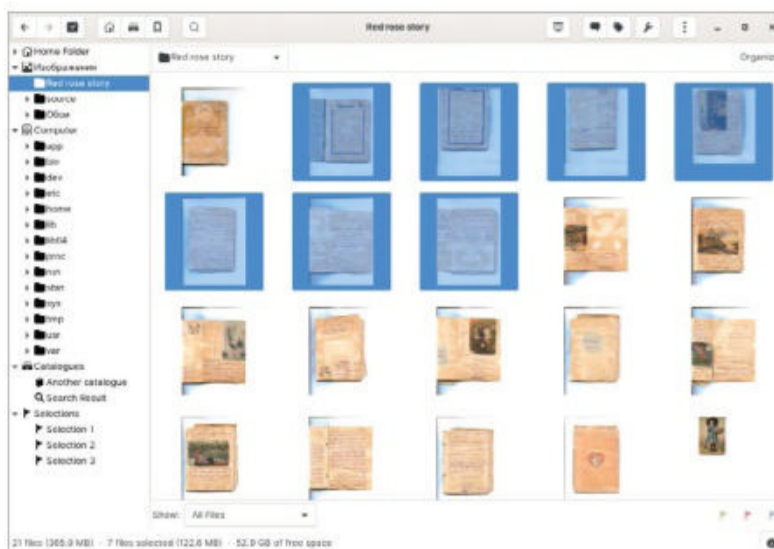
gThumb

Version: 3.12 Web: <https://github.com/GNOME/gthumb>

People who organise their image libraries using open source software know how difficult it is to find the perfect piece of software that can match all their needs. Traditionally, we've seen a years-long competition between *digiKam* and *Shotwell* in that field, but sometimes you might need to have a lighter alternative at hand.

Perhaps that's where *gThumb*, a GTK-based image viewer and organiser, can help. Lately it saw another update, which brought some amazing new features, so let's see what version 3.12 of *gThumb* offers. The most-anticipated addition is support for modern image formats, such as HEIC, HEIF, AVIF and JXL (from the all-new *JPEG XL* that we cover as another pick below). You can now confidently view, edit, organise and export images using those next-generation formats. All of them assume smaller file sizes with a negligible loss in quality.

We like *gThumb* for its visual clarity combined with a solid set of features. For most of us, *gThumb* could easily replace *Shotwell* as a daily photo library manager. The tool can assign tags and perform search queries



gThumb is lightning-fast and packed with most of the things you'd expect from a decent photo library manager.

based on those tags, and it also can do some sophisticated filtering.

For instance, imagine you have a huge photo library made of thousands of shots, from which we need to filter out a select number based on certain criteria: file size, date, format, description or whatnot. *gThumb* supports advanced filtering from any location, no matter whether it is a convenient directory tree, catalogues or selections. In *gThumb*, catalogues work like albums – you can add any image to a catalogue via its context menu. Selections provide a light table-like feature, in case you need to quickly review images from different catalogues. *gThumb* can do basic editing, too, including rotation, colour and gamma levels, curves, configurable unsharp mask, red-eye removal and more. All of these frequently used tools conveniently appear when you hit the Edit button on *gThumb*'s header bar.

IMAGE ENCODER

JPEG XL

Version: 0.5

Web: <https://github.com/libjxl/libjxl>

Back in **LXF205** and **LXF222**, we admired FLIF as a cool, smart replacement for lossless PNG. Although opening a FLIF image took substantially longer than a PNG, you still got a decent benefit in size reduction of around 15 to 20 per cent. FLIF has been declared obsolete, officially replaced by the far more advanced encoder *JPEG XL*. It combines several modes under one roof in such a way that you can convert your existing JPEGs, PNGs and GIFs into a single target format (JXL) regardless of their type. As such, *JPEG XL* knows how to handle photos, drawn artwork, screenshots and other sorts of synthetic imagery. You can use the standard CLI tools for encoding (`$ cjxl`) and decoding (`$ djxl`) your files, plus equip your *Qt5* plugins installation with the unofficial JXL support plugin (<http://bit.ly/3zVJPIQ>).

Let's see how *JPEG XL* performs. A plain conversion of a PNG file using defaults looks as simple as `$ cjxl input.png output.jxl`

In this case, the encoder applies the VarDCT compression algorithm, which delivers visually lossless



This next-gen format could eventually edge out JPG, PNG and WebP, which have seemingly been around forever.

results with awesome file size reduction. Decoding such files back to PNG with `djxl` produces an almost same-sized yet not identical image. The default encode behaviour when dealing with JPGs is different. Plain conversion applies the legacy file repacking and yields 15 to 20 per cent file size reduction without introducing further quality loss to an already lossy JPG. As such, decoding it back results in an identical JPG.

The VarDCT method is far superior to the legacy one (the latter is comparable with *libjpeg-turbo*, which is good but not spectacular). To get the most out of *JPEG XL* when dealing with JPGs, convert the source image to PNG, then repack it into the target JXL file. That's for (nearly) lossless mode – you can alter quality and distance options for `cjxl` and further reduce the file size.

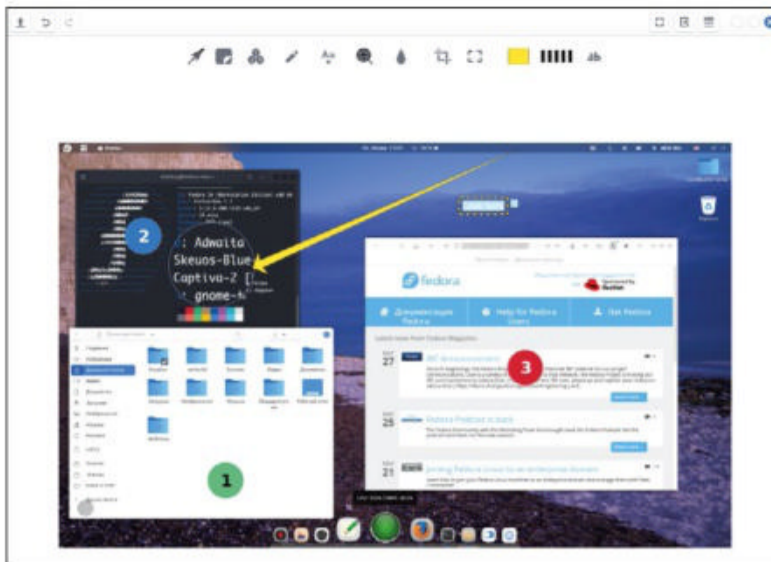
IMAGE ANNOTATION TOOL

Annotator

Version: 1.0 Web: <https://github.com/phase1geo/Annotator>

The world of Linux applications is known to have a massive number of small accessories and handy utilities for almost every task. Our quest this time is to edit screenshots with a brand new and already very capable tool called *Annotator*. This application joins a family of existing tools such as *Ksnip*, *Flameshot* and *Shutter*, which you may already use. *Annotator*, however, can't take screenshots itself; it focuses on annotating – adding elements to original images. In that sense, *Annotator* is an advanced image viewer, or special-purpose editor, if you wish. The application enables you to open any valid image file or paste the contents of the clipboard (provided it contains a screenshot or manually copied image).

The main benefit of using *Annotator* is that its impressive set of tools is nicely arranged and easily available. For example, with *Annotator* you can add arrows, boxes, shapes, counter badges and many other sorts of visual highlights. It is also possible to obfuscate sensitive data by blurring certain areas using the Blur Box tool, emphasise areas with Magnifier and, of course,



Rich annotating tools mean you can create much fancier screenshots.

play with text captions and enjoy rich text editing. Whatever you decide to add to an image stays editable and movable. Any added shape (except for fancy stickers) is a vector object, which you can drag or resize. You can change the nodes of complex shapes such as arrows and transform them into other shapes.

While *Annotator* is a new program, it already rivals its more heavyweight competitors. The magnifier tool is an *Annotator*-exclusive, and in our view it's a gorgeous effect that will be useful to a great many people. The interface is easy on the eye and well thought-out, with everything in the right place. The floating text edit panel makes it super-easy to format headers, code blocks, sub- or superscripts, and the Export Image menu offers quick export options, including copying the edited image back to the clipboard.

SYSTEM MONITOR

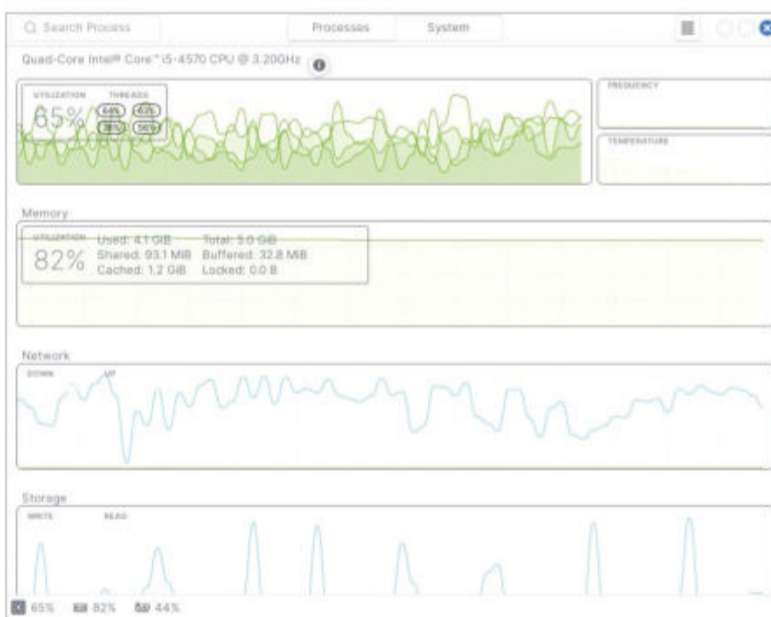
Monitor

Version: 0.10

Web: <https://github.com/stsdc/monitor>

This little utility was initially designed to fix the problem of a missing system resource monitor in Elementary OS. Even though you could always use command-line tools such as *top*/*htop*, there was no dedicated graphical tool for the job. Thanks to *Monitor*, you now have a much better insight into resource usage, CPU load and other related metrics. Sure, the software isn't unique – we already have *Gnome System Monitor* – and at first glance *Monitor* seems like just another GTK-based program with the same feature set, but that's not quite true. *Monitor* has a substantially different look and feel, which changes the way we view system resource use.

Monitor has two main tabs: one for system processes and another for resources usage. When exploring the processes list, you can sort them by name, PID number, CPU or memory usage, and thus quickly track down the required string out of the usually long list of suspects. However, you can also instantly see which files have been opened by a particular process. In *Monitor*, you don't need to open a process properties window or do



Monitor combines spectacular looks with an impressive feature set.

anything else to see its files – all the relevant data is always at hand.

The System tab has the usual graphs showing CPU, memory and network utilisation, plus a fourth graph dedicated to disk throughput. The latter is perfect for tracking intense read and write disk operations that can cause the whole OS to become slow and unresponsive. As for the first two graphs, click their Utilisation boxes to reveal extra details, such as threads for CPU and the cached/buffered/shared figure for memory. Further exploration of the interface reveals lots of cool extra details. For instance, if your system is set up to monitor CPU temperature via *lm-sensors*, *Monitor* displays relevant figures next to the CPU graph. If you don't have a GUI system monitor, this tool is highly recommended.

PINBALL GAME

SpaceCadetPinball

Version: 1.1.1

Web: <https://github.com/k4zmu2a/SpaceCadetPinball>

We've never had the chance to feature the glorious *SpaceCadet* in the past because it was a closed-source proprietary game. Even though it took very little effort to extract the demo version of the game from a Windows installation and run it via *Wine*, it didn't fit in with the type of applications we normally highlight here in *HotPicks*.

However, there's now a new GitHub project that implements the native SDL2 port of the game, and the (pinball) tables are turned. *SpaceCadetPinball* is a small project that enables you to play the same-named arcade pinball game as a fully native Linux application. Just as with many other projects of this kind, you do need to have an original copy of *SpaceCadet* in order to use the artwork and other game assets within the open source *SpaceCadetPinball* loader.

As a quick reminder, the version that shipped with Windows was a truncated demo, featuring just one table and low-resolution graphics. It was basically an ad campaign for *Full Tilt! Pinball*, which was originally developed by Cinematronics and published by Maxis

Software. The full version boasted three tables and better artwork, which was particularly noticeable when playing the game in full-screen mode. Anyway, whatever version you choose to go with, they all have definitely long become abandonware.

Right now, *SpaceCadetPinball* only supports the first table, which sees you at the control panel of a fictional space ship. The table field is generously populated with bumpers, switchers, ramps, gates and whatnot to back up the story of a brave explorer travelling on his ship and fighting space aliens. Even though all user interaction is limited to twitching those two lower flippers, you can do a lot of things in the game: take missions, complete tasks and advance your cadet career. All very entertaining.

In order to run the game, you need to compile the *SpaceCadetPinball* executable with *cmake*, place the resulting file in the same directory as the game's DAT file, and run it.



Complete three missions and get promoted to the next rank. Be careful not to miss the ball between those flippers...

CITY TRAFFIC SIMULATOR

A/B Street

Version: 0.2.61

Web: <https://github.com/a-b-street/abstreet>

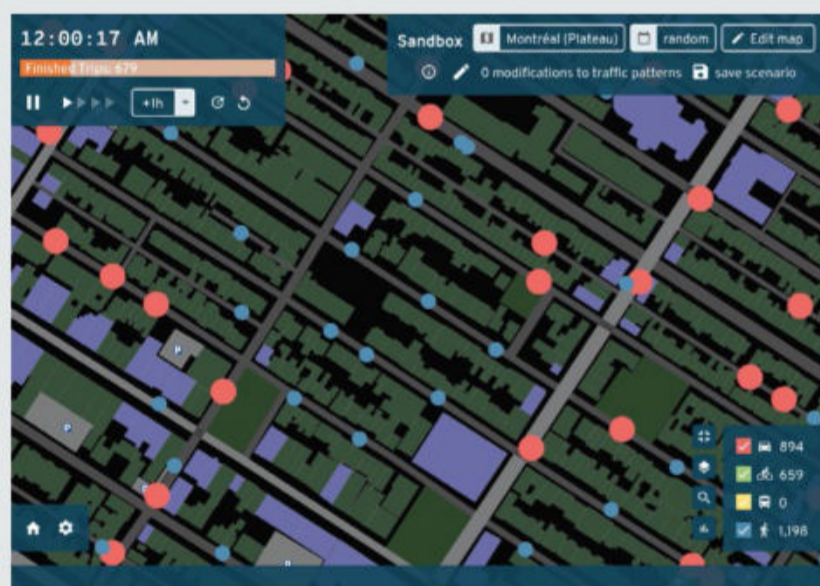
For as long as we can remember, there has never been a decent *SimCity* clone for Linux. So, if you wanted to try out the role of city mayor, you needed to look elsewhere. However, we've discovered a city simulation game that brings a completely new experience. *A/B Street* is an open source city traffic manager that puts you in control of real-world transportation issues. Moreover, *A/B Street* can import maps from OpenStreetMaps, enabling you to manage a real city's routes in a sandbox.

The most impressive thing about *A/B Street* is that its mechanics are incredibly close to what goes on in real life and what genuine city managers do regarding public transport. In the game, you have a top-down view over a city, as though you're flying around it using a drone. There are streets, lanes, junctions, pavements and all other elements of urban routes. You can see cars, buses and bikes travelling along the roads, pedestrians walking around, and more. The goal is to solve common problems and optimise the transport network to create shorter journeys and a safer environment. The game

greet you with a well-made tutorial that explains all the

essentials and eventually asks you to complete small tasks, such as altering lane restrictions in order to make the journey safer or save time.

The zoomed-out city plan doesn't show a great deal of detail, but once you zoom in you can interact with a whole range of elements, including all the lanes on a road, various buildings and smaller routes. For instance, it's easy to make changes to a junction: select the junction in Edit Map mode, make and apply your changes, go back to the map, then press the Spawn Agents Here button to see how cars, bikes and people now use the junction. *A/B Street* can be used to solve issues that could lead to traffic jams, or eliminate risks for people without cars. It also offers help, such as not letting you close a junction without reconnecting pavements in some other way.



A/B Street has lots of real-world city maps, which you can test by generating random transport routes or making custom rides.

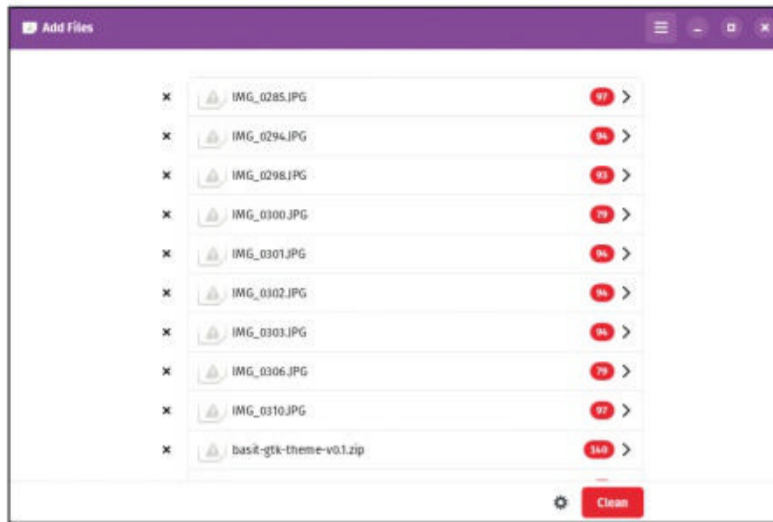
PRIVACY TOOL

Metadata Cleaner

Version: 2.0.1 Web: <https://gitlab.com/rmnvgr/metadata-cleaner>

Privacy has long been a concern among Linux users, because the more we use our computers, the more data is collected and received by third parties. Of course, there are many aspects to security, but here we're focusing on just one: the traces we leave within our files. It's no secret that images, music and video files contain lots of metadata to help us obtain information to better identify the file. For example, photo files can store information about date, time, shutter speed, camera model and more. Sometimes we don't want all this data to be made public, either for the sake of privacy or to prevent confusion. *Metadata Cleaner* is a small GTK-based application dedicated to this worthy cause.

It's very simple to use: launch the tool, hit the Add Files button and select something from your file system. *Metadata Cleaner* examines the files and displays them



Many people don't realise how much information their files might be giving away.

as a list. Each entry bears a red badge displaying the number of strings found; an average photo taken by a smartphone usually contains over 90 fields. Click it to open a list of them all. You're not supposed to edit any of the metadata, but if you hit the red Clean button, everything is wiped away. *Metadata Cleaner* alters a file's contents but shouldn't break anything.

Bear in mind that while the absence of any metadata is perfect for privacy, it does prevent good file indexing and sorting in various catalogue and library managers, such as *digikam*, *Shotwell* and so on. If unsure, make a backup of the original files; that's exactly what *Metadata Cleaner* politely asks you to do when running it for the first time. Alternatively, enable the light cleaning mode in the program's settings. That way, *Metadata Cleaner* enables several sorts of embedded metadata to remain.

WEBCAM

Linux Fake Background Webcam

Version: GIT Web: <https://github.com/fangfufu/Linux-Fake-Background-Webcam>

There are mixed views about the effects of homeworking on employee productivity during this challenging period (*clearly not at LXF Towers-ED*), but one undisputed result is an increase in the use of video calls and group video conferencing. This *HotPick* adds some extra functionality to your webcam. Its full name is *Linux Fake Background Webcam (LFBW)*, which is fairly self-explanatory.

The idea is to replace your real background with a fake one, using a bunch of open source tools. You need a V4L2-compatible video device (which could be an Android phone), a relatively fresh software stack, including recent versions of Python3 and OpenCV, and enough resolve to set up the `v4l2loopback` kernel module. The latter is the main obstacle, as some distros don't include it by default, forcing you to build it from sources (<http://github.com/umlaeute/v4l2loopback>). After building and installing the module, ensure it's



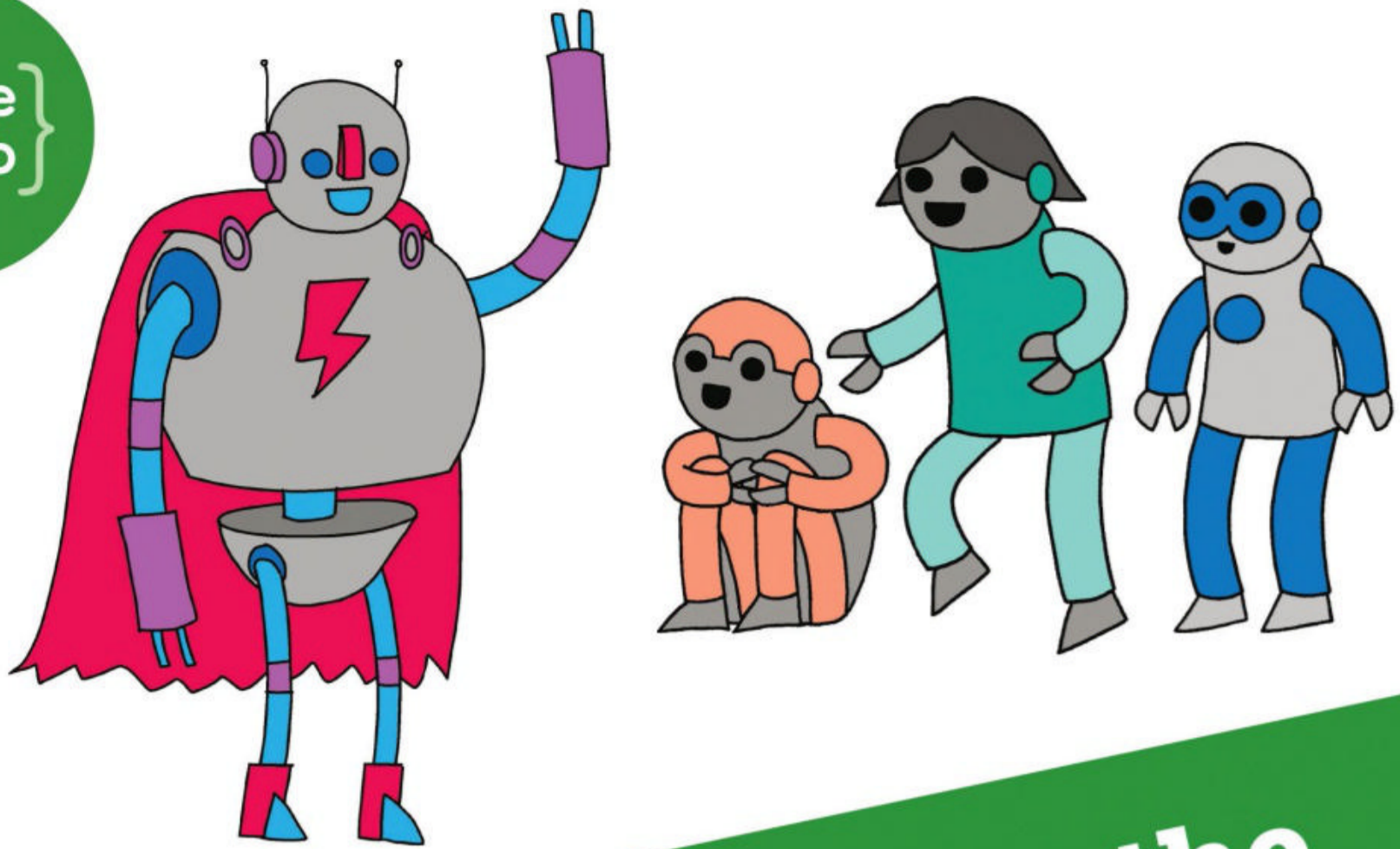
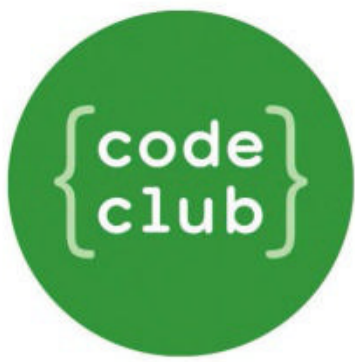
loaded (`$ sudo modprobe v4l2loopback`). The module creates the extra `/dev/videoX` device, which we'll be using as a fake webcam. Start *LFBW* like this:

```
$ python3 fake.py -W 1280 -H 720 -w /dev/video1 -v /dev/video0
```

Adjust the command with your webcam resolution and the correct names of source and target devices. Here, `video1` is the real webcam device (source), while `video0` is the output fake camera (target).

LFBW takes the video stream from the real webcam device, removes the still background using some OpenCV magic (it's best to sit before a bright wall) and replaces it with whatever you defined either directly in the command, or in the config file (see `config-example.ini`). Without altering the defaults, the tool takes you to a lectern in the White House. To change the scene, you need a new background image, and a file for masking fictional foreground objects. **LXF**

Talk to the world from a lectern, beach or anywhere else you have a picture of...



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

PYTHON

Using Python sockets for multiplayer gaming

Discover how to implement multi-player gaming in a Galaxian-style shooter with **Andrew Smith**, in the second part of coding Star-Fighter!



OUR EXPERT

Andrew Smith is a software developer for NHS Digital, has a bachelors degree in software engineering and a master's degree in computer networks.

This month we're going to look at the multiplayer version of *Star-Fighter* that was covered in **LXF282** and played as a single player game (created by Francis Michael Tayag). We'll cover the network programming techniques that make it possible to play this game across a LAN (local area network) and where in the source code the changes have been made.

Both instances of the game (server and client) will be executed on the same machine, even though it's possible to run them on separate machines on the same LAN. Network programming in Python is an advanced topic so if any readers are new to Python, it may be advisable to focus on just the setup and execution of the project. It's also worth pointing out that this tutorial won't cover how parts of the game work or how the game is structured, because this was discussed in **LXF282**. Instead, this tutorial focuses on the networking element of the game (which uses the sockets library) and where changes have been made in the game to make multiplayer action possible.

We're aiming to run two instances of the game on the same device. It's possible that some readers may not have two computers readily available, so for this tutorial just one device is used. Let's begin by setting up our Python development environment.

Installation and setup

To install Python, open a terminal window (Ctrl-Alt-T) and type `sudo apt-get python3` followed by `sudo apt-get install pip3`. Then install the PyGame module by typing `pip3 install pygame`. To make sure that you're using PyGame version 2.0 (*Star-Fighter* does use a recent version of PyGame), type `python3 -m pip install pygame==2.0.0`. You should now have version 2.0 of PyGame installed. Type the following to check what you've just installed:

```
python3
import pygame
quit()
```

If all's well then your screen should look the same as the screengrab (facing page, top right).

Finally, grab a copy of the *Star-Fighter* project source code by cloning the GitHub repository. Before typing the



An example of multiplayer gameplay. Shown are both the server and client instance of the program running on the same device.

following to clone (copy) the GitHub repository, move into a folder on your system that you'd like the project to be copied to.

```
git clone https://github.com/asmith1979/starfighter_multiplayer/
```

As an example, the whole project has been put into a folder called **PythonProjects**, which was created before downloading the project.

To edit and view the source code you can either use a basic text editor or something more specific such as *Notepad++*, *PyCharm* or *VS Code*. In this tutorial, we'll be using *gedit* to view and edit the source files. If and when using this method to view or edit source files, it may be helpful to open up three console windows: one for editing/viewing source files and the other two terminal windows for executing the PyGame code.

Star-Fighter in multiplayer mode

For the multiplayer feature of the game to work on the same device, you'll need to modify the script file **game.py**. Open up **game.py** in your chosen editor and find the property called **multiplayerDemo** – there should be two instances of this. The value should be currently set to **False**. Change both instances of **multiplayerDemo** to **True**, and then save.

QUICK TIP

You may find it useful to use three console windows for this tutorial: one for editing or viewing source code; one for running the server instance; and one for running the client instance.

On both of the open terminals, run *Star-Fighter* by typing the following into each terminal window:

```
python3 game.py
```

The game's music may sound odd with both instances of the game playing it at the same time on the same device. So on both instances of the game, open the Options menu and select the Sound menu. Once there, set the music volume at 0 and then go back into the Options menu. No more distracting music!

Out of the two instances running the game program, one will need to run as the game's server and the other instance will need to run as the client player instance. In each of the instances, go into the Multiplayer menu settings via the Options menu.

When you first access the Multiplayer menu, the default mode that's selected is standalone, which means the game is currently played in single player mode. Pressing the right cursor key selects server mode so the game instance will act as a server. Pressing the right cursor key again puts the game into client mode.

Choose one of the instances to be in server mode and the other game instance to be in client mode. Because both instances are now running with `multiplayerDemo=True`, this acts as an override to using the real IP addresses and both instances use the local host address, 127.0.0.1 instead of the actual IP address of the device and both use different port numbers. The server instance uses port number 20001 and the client instance uses port number 20002.

Once the server mode on the server instance of the game has been selected, choose Accept and Close, and then go back to the main menu and start the game.

In the game instance of the client, accept and go back to the main menu and start the game. This should display a message on the screen that the client has been connected to the server, and the game will begin.

You can play both server and client individually by clicking the executing game instance. Observe the effect of this on the other game instance and vice versa.

Converting to multiplayer

Before any network code could be implemented or used in the *Star-Fighter* project, various parts of the program had to be altered for the game to accept there was the

```
andrew@dell-ubuntu-01:~$ python3
Python 3.8.10 (default, Jun 2 2021, 10:49:15)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import pygame
pygame 2.0.0 (SDL 2.0.12, python 3.8.10)
Hello from the pygame community. https://www.pygame.org/contribute.html
>>> quit()
andrew@dell-ubuntu-01:~$
```

possibility of being more than one player and if there was more than one player, to handle the extra player equally. The game code has been adapted to handle up to two players at a time.

In `scenes.py` in GameScene class constructor type

```
self.playerCollection = []

# Initialize the player
self.player = Player(P_LAYER_IMGS, BULLET_IMG,
self.P_Prefs)

self.playerCollection.append(self.player)

# Add second player to screen if multiplayer version is
selected
if self.P_Prefs.multiplayer == SERVER_MODE or self.P_
Prefs.multiplayer == CLIENT_MODE:
    self.playerTwo = Player(P_LAYER_IMGS, BULLET_
IMG, self.P_Prefs)
    self.playerTwo.isMultiplayer = True
    all_sprites_g.add(self.playerTwo)
    self.playerCollection.append(self.playerTwo)

all_sprites_g.add(self.player)

# Create a spawner
self.spawner = Spawner(self.playerCollection, self.g_
diff, self.P_Prefs.multiplayer)
```

A second player is only created when the game program is either in SERVER_MODE or CLIENT_MODE (a mode of network play) which is selected by the player in the multiplayer menu. If the mode is left as default, STANDALONE, the game continues as a single player game. Note that a property has been added to the

Here's the PyGame version installed. You should see that version 2.0 of PyGame has been installed.

» IP ADDRESSES AND LOCALHOST

An IP (internet protocol) address identifies a device on the internet or on a local area network (LAN) if an internet connection exists. An IP address can be identified by four numbers separated by three full-stops. For example a valid IP address could be 172.168.1.212. A local host address is always referred to as 127.0.0.1 and is also known as the loopback address.

In Multiplayer Demonstration mode, which this tutorial is based on, only local host addresses are used in both instances so both use 127.0.0.1 with different port numbers so that the communications don't clash on the same

device. When the Multiplayer Demonstration mode isn't activated, the IP address of the devices is used and each instance has to run on a separate device for the multiplayer feature of the game to work. You'll notice that there are some multiplayer settings defined in `games.py` that refer to IP addresses.

```
self.clientAddressPort = (['127.0.0.1',
20002])
self.serverAddressPort = (['127.0.0.1',
20001])
self.clientIPAddress = self.
clientAddressPort[0]
self.serverIPAddress = self.
serverAddressPort[0]
```

```
self.clientPort = 20002
self.serverPort = 20001
self.multiplayerDemo = False
```

By default, both client and server IP addresses are set up as the localhost address (127.0.0.1) because at this stage in the program there's nothing to identify otherwise. An address port variable is also set up for both server and client that contains both the IP address and port number being used.

The port numbers in this example are used to receive communications. Note that 20001 is to receive data on the server port and port number 20002 is to receive data on the client port.

Player class to identify if the player is multi-player identified by `isMultiplayer`, which is a Boolean variable.

Furthermore, functions throughout the game program that expect an instance of a player to be passed in are passed in as a collection of players. This avoids having to add another argument to a function to take an extra player. You'll see in the previous source code example that `self.playerCollection` is used to hold a collection of players in the game.

When playing in multi-player mode, the server is the side that has dominant control, because the scores and both players' health are fed from the server to the client. The position of the client spaceship is the only property passed from the client to the server, which the server then inserts into its current "game model" instance.

At the end of the GameScene constructor, the main communications points are set up so that gameplay data can be transferred between server and client instances of the game program:

```
# Create both server and client stub
self.serverStub =
MultiplayerDataTransferServer(self.P_Prefs)
self.clientStub = MultiplayerDataTransferClient(self.P_Prefs)

# If acting as server, setup server listening thread
if self.P_Prefs.multiplayer == SERVER_MODE:
    self.threadedServerProcess = threading.
Thread(target=self.serverStub.setupServerListening)
    self.threadedServerProcess.start()

# If acting as client, setup client listening thread
if self.P_Prefs.multiplayer == CLIENT_MODE:
    self.threadedClientProcess = threading.
Thread(target=self.clientStub.clientEndPoint)
    self.threadedClientProcess.start()
```

Even though both server and client points are set up in the constructor, only one is used depending on the game instance being played by the player. If the game



Here's the Server player waiting for Client player to connect. Remember to always start the server before the client.

instance running is acting as the server, the server stub will be used. If the game instance is running as the client, the client stub will be used. On each instance of the game, there's a thread that's used to listen for incoming communication and to also send that data out again. Threads enable the program to flow naturally while another process can execute alongside it.

Server-side processing

`MultiplayerDataTransferServer` is the class used for all server-side processing. Once a player of the game program acts as the server in the multiplayer menu, this is the class that manages the communication between the server instance of the program and the client instance. The server class contains the following functions (aside from the constructor):

- `getServerIPAddress()`
- `clientInList()`
- `clientLimitReached()`
- `getConnectionRequest()`
- `sendDataToClient()`
- `setupServerListening()`
- `selfTermination()`

The main functions of interest out of this class are `setupServerListening()`, `sendDataToClient()` and `selfTermination()`. The other functions in the class can be regarded as helper functions.

The function, `setupServerListening()`, is the main function of interest in the class because it first sets up a socket and continually listens for incoming communications from the client instance of the program. This is also the function that's threaded on the server side. The function `sendDataToClient()` is used to send game data to client, which includes data for both players as well as game data such as the score. The function `selfTermination` is used to end the server process at the end of the multiplayer game session.

Client side processing

`MultiplayerDataTransferClient` is the class used for all client-side processing and is located in `scenes.py`. The main functions of interest in this class include:

- > `clientEndPoint()`
- > `sendDataToServer()`
- > `selfTermination()`

The other functions present in this class are used as helper-functions. In this class, `clientEndPoint()` is the function that's threaded and is used to receive communication from the server side of the game.

Transferring game data

The transfer of player/game data to and from the client and server is done in the update function of the GameScene class located in `scenes.py`. Data for both the client and server side instances are dealt with here.

There is a class that has been created and used called PlayerData located in `sprites.py` that holds mainly player data, but also some game data. It's a collection of this data that's passed between client and server. The example below shows that the data from the server is processed by the client once it's received.

```
if self.P_Prefs.multiplayer == CLIENT_MODE: # Client mode
```

```

# Data Transfer from Server (To Client)
if len(self.clientStub.objectCollection) > 1:
    self.clientStub.dataTransferInProgress = True
# Player Two Data (Remote player)
playerData = self.clientStub.objectCollection[1]

self.playerTwo.setData(playerData)

# Get general game data
self.score = playerData.gameScore
self.is_gg = playerData.gameEnd

# Player One Data (Client Player)
clientPlayerData = self.clientStub.
objectCollection[2]
self.player.health = clientPlayerData.health
self.player.isDead = clientPlayerData.isDead
self.player.gun_level = clientPlayerData.gun_
level
self.player.prev_gunlv = clientPlayerData.prev_
gunlv
....

```

From studying this code, you'll see that a variable called `dataTransferInProgress` is set to True near the start. This is to tell `clientEndPoint()` that no more data is to be collected until the data has been fully taken from the data structures and put into the game model. Once this is done, it's set to False and the data from the server is then continued to be retrieved. The data is transmitted in a layered way and then processed in a layered way. The first data structure (data layer 0) is used for general information purpose such as connection requests or requests to terminate a service. The following two structures (data layer 1 and 2) that are passed contain player and game data.

The same is also done for the server-side:

```

if self.P_Prefs.multiplayer == SERVER_MODE: # Server-
Mode

# Data Transfer to Client (From Server)
# Clear send object collection to reset
self.sendObjectCollection.clear()
sendGameDataObject = MultiplayerMessage('GAME
DATAFROMSERVER', '0.0.0.0', 0)
self.sendObjectCollection.
append(sendGameDataObject)

# Add server data
pData = PlayerData()

# Add Server Player Data
pData = self.player.getData()

# Add Server Game Data
pData.gameScore = self.score
pData.gameEnd = self.is_gg
....

```

From studying these two sections of code in `scenes.py`, you'll learn that as well as retrieving data, there's also a side of the process that collects data to send back to the other side. The server will collect data from its own game instance, then send it to the client. The

» THREADS IN PYTHON

A thread is a separate flow of execution that enables a process to operate alongside the main process of an application or in the case of this project, a video game. To start using the Threading functionality in Python, the threading library needs to be imported.

In the Multiplayer *Star-Fighter* project, threads have been set up and used to manage communication between the client and server instances of the game program. At the same time the game can continue as normal.

The first step in using a thread is to set one up. This is done like so:

```

self.threadedClientProcess = threading.Thread(target=self.
clientStub.clientEndPoint)
self.threadedClientProcess.start()

```

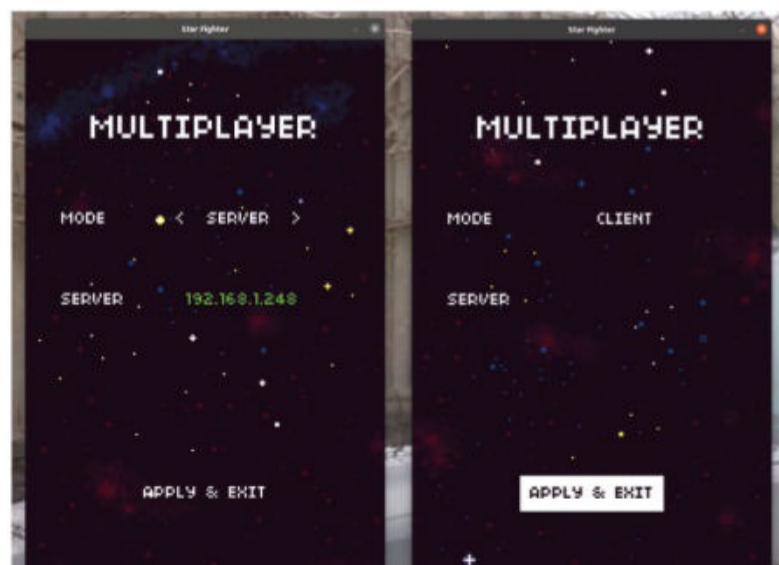
As can be seen from the above code segment, the thread is set up to target a function to process as the thread. In this case it's a function called `clientEndPoint()` that will be used as the thread. This function can be found in the `MultiplayerDataTransferClient` class located in `scenes.py`. In addition, threaded functions are usually looped processes so it's important to fully define and be clear on the end condition that will end that process, then end the thread itself. In `clientEndPoint()` there's a Boolean condition that's used to end the looping process

client instance will collect the relevant data (spaceship position) to send back to the server for processing.

The original game has been further modified than what's been shown in this tutorial to incorporate a second player, but it's beyond the scope of this article to cover this aspect. We're only focusing on the network coding elements of the game. Further aspects that could be worked on to improve this project include:

- An improved algorithm to generate enemy game characters (located in `spawner.py`)
- An improved algorithm to generate power-ups (located in `spawner.py`)
- Better client authentication (only basic client IP authentication has been implemented)
- Adapt code further so that it can be played over the internet instead of just via a LAN

Have fun with the code and let us know what improvements you create! And remember – when looking for a function or variable name in a script, you can use the IDE's search facility to go directly to it instead of scrolling through all the source code. **LXF**



The multiplayer menu. Select to either act as the server or have the client connect to the server.

» **GET INTO A KILLING FRENZY AND** Subscribe now at <http://bit.ly/LinuxFormat>

REST

How to develop a RESTful client in Go

Part Two!
Did you miss part one? See page 66 to get hold of it!

Learn how to develop a command-line client to access a RESTful server in Go with **Mihalis Tsoukalos** and the Cobra library.



OUR EXPERT

Mihalis Tsoukalos is a systems engineer and technical writer. He's the author of *Go Systems Programming and Mastering Go*. You can reach him at [@mactsouk](https://twitter.com/mactsouk).

The subject of this month's tutorial is RESTful services. In particular, you're going to learn how to develop a client for an existing RESTful server using the Go programming language. The client is going to be able to communicate with the RESTful server that we developed in part one (see **LXF282**), which means that the command line utility must support all server endpoints. Let's get started.

The supported REST API

First, let's recap what our REST API client needs to support. The REST API is defined on the server side, but remember that you don't have to support all endpoints. What is important is to use the correct endpoints along with the expected HTTP methods and JSON records. Once again, we need to marshal and unmarshal JSON data when interacting with the RESTful server. However, we're not going to discuss the JSON marshalling and unmarshalling process in this tutorial – if you don't know how to do this, get hold of the previous tutorial.

As a reminder, the REST API that we need to support includes four endpoints. These endpoints are **/time** (GET HTTP method), **/insert** (POST HTTP method), **/delete** (DELETE HTTP method) and **/list** (GET HTTP method). Only **/insert** requires JSON input whereas **/list** returns an array of JSON records instead of single JSON records. Additionally, **/delete** requires user input to know the username that's going to be deleted, if it can be found in the server "database".

Utility structure

We're using *cobra*, which is a Go package that's not part of the standard Go library. The development of the command line utility is going to take place under **~/go/src**. More specifically, under **~/go/src/restclient**. First of all, we have to create the structure of the command line utility using the **~/go/bin/cobra** utility:

```
$ cd ~/go/src/restclient
$ ~/go/bin/cobra init --pkg-name restclient
$ ~/go/bin/cobra add time
$ ~/go/bin/cobra add insert
$ ~/go/bin/cobra add list
$ ~/go/bin/cobra add delete
```

```
restclient go mod init
go: creating new go.mod: module restclient
go: to add module requirements and sums:
  go mod tidy
restclient go mod tidy
go: finding module for package github.com/spf13/cobra
go: finding module for package github.com/spf13/viper
go: downloading github.com/spf13/viper v1.8.1
go: downloading github.com/spf13/cobra v1.2.1
go: found github.com/spf13/cobra in github.com/spf13/cobra v1.2.1
go: found github.com/spf13/viper in github.com/spf13/viper v1.8.1
go: downloading github.com/fsnotify/fsnotify v1.4.9
go: downloading github.com/hashicorp/hcl v1.0.0
go: downloading github.com/magiconair/properties v1.8.5
go: downloading github.com/mitchellh/mapstructure v1.4.1
go: downloading github.com/pelletier/go-toml v1.9.3
go: downloading github.com/spf13/afero v1.6.0
go: downloading github.com/spf13/cast v1.3.1
go: downloading github.com/spf13/jwalterweatherman v1.1.0
go: downloading github.com/spf13/pflag v1.0.5
go: downloading github.com/subosito/astenv v1.2.0
```

This is how you build the cobra project for the first time. This includes enabling Go modules using "go mod init" and "go mod tidy".

The second command is for initialising the *cobra* project. The remaining commands are for creating the desired utility commands. So, the *restclient* utility needs to support the **insert**, **delete**, **list** and **time** commands. After these *cobra* commands, the structure and the files of the utility, as presented by *tree*, are going to be as follows:

```
tree
.
├── LICENSE
├── cmd
│   ├── delete.go
│   ├── insert.go
│   ├── list.go
│   ├── root.go
│   ├── time.go
└── main.go
```

1 directory, 7 files

All these Go source files contain lots of comments and a default help message that you can delete. However, if you're not comfortable with Go code, you can skip that step because it's not necessary. One way or another, you don't have to touch **main.go**.

At this point, it would be good to try and see whether the utility works or not. The process is shown in the screenshot (above). Note that you need to run **go mod init** and **go mod tidy** only once and that you're allowed to delete **go.mod** and **go.sum** and recreate them any

QUICK TIP

If you want your programs to validate JSON input according to some user-specified rules, check out the Go validator package at <https://github.com/go-playground/validator>.

time you want. Additionally, all lines beginning with `go:` are related to Go modules and will appear only once. The final lines of the output present the default help message of the `cobra` project that you can modify by editing the respective source code files.

Down to the root.go

To add global variable and command line flags, we need to modify `root.go`. First we begin by declaring the following four global variables in `./cmd/root.go`:

```
var SERVER string
var PORT string
var username string
var password string
```

Have in mind that all `username`, `password`, `SERVER` and `PORT` variables must be defined in `./cmd/root.go` to be accessible by all the source files of the project. All of these are going to be used in a while.

After that we should define the command line flags. For reasons of simplicity we're going to define two global command line flags for passing the `username` and `password` values. Each command is only going to look for what's required for the endpoint that's visiting. Simply put, only `insert` requires valid values for both of them and only `delete` requires a valid value for the `username`. All other command ignore them even.

Finally, we need two more global command line flags for specifying the RESTful server and the port number the server listens to. These two flags are going to be used by all commands. The implementation of each command needs to access and use the global variables that store the user values passed to the utility via flags.

The following code presents the definition of the supported command line flags in `root.go`:

```
rootCmd.PersistentFlags().StringVarP(&username,
"username", "u", "default", "The username")
rootCmd.PersistentFlags().StringVarP(&password,
"password", "p", "default", "The password")
rootCmd.PersistentFlags().StringVarP(&SERVER,
"server", "s", "localhost", "RESTful server")
rootCmd.PersistentFlags().StringVarP(&PORT, "port",
"P", ":1234", "TCP Port")
```

Due to the use of `StringVarP()`, all values are going to be stored as Go strings. Additionally, the values of the command-line flags are automatically stored in the variable names that are passed as the first parameter to `rootCmd.PersistentFlags().StringVarP()`. So, for the

```
16 var timeCmd = &cobra.Command{
17   Use:   "time",
18   Short: "The time command",
19   Long:  "The time command asks the RESTful server for the time.",
20   Run:   func(cmd *cobra.Command, args []string) {
21     req, err := http.NewRequest(http.MethodGet, "http://"+SERVER+PORT+"/time", nil)
22     if err != nil {
23       fmt.Println("Timefunction - Error in req: ", err)
24       return
25     }
26
27     c := &http.Client{
28       Timeout: 15 * time.Second,
29     }
30
31     resp, err := c.Do(req)
32     if err != nil {
33       fmt.Println(err)
34     }
35   }
```

This is the implementation of the `time` command as found in `./cmd/time.go`. The structure of the remaining commands is similar to the one shown here.

first statement, we have the following: the name of the variable, which is passed as a pointer, which will be used for storing the value `"username"`, which in this case is the same as the name of the command line flag (`--username`). That flag has a shortcut named `-u`, a default value of `"default"` and a description of `"The username"`. All other statements are analogous to this first one. All these flags have default values.

Next, we should begin implementing the commands, one by one, beginning with the `time` command.

Implementing time

The main reason that we start with the `time` command is that its implementation is the simplest of all, because it doesn't send any data to the server and it receives the server response as plain text. However, most of the code found in `./cmd/time.go` is also going to be used in the other commands, because the way the commands interact with the RESTful server follows a specific pattern. First, we read the necessary parameters given as command line arguments, then we construct the client request and finally, we read the server response.

The most important part of `./cmd/time.go` is `req, err := http.NewRequest(http.MethodGet, "http://"+SERVER+PORT+"/time", nil)`

This is where we construct the URL that we're going to visit. As mentioned earlier, the same technique is used in all other commands. The last part of the URL is the name of the REST API endpoint (`"/time"`). What's important here is the use of `http.MethodGet` as the first parameter of `http.NewRequest()` to declare the HTTP method that's going to be used (GET) – this is a important detail because, as you might recall, each endpoint requires a specific HTTP method to work.

» A FEW WORDS ABOUT COBRA

Go is particularly effective at developing command line utilities; `docker` and `kubectl` are two famous utilities developed in Go. Both of them were created using the `cobra` package.

Commands in `cobra` can have one or more aliases, which is very handy when you want to cater for both amateur and experienced users. Additionally, `cobra` supports persistent flags and local flags, which are flags that are available to all commands; and flags that are available to given commands only, respectively.

By default, `cobra` uses `viper` (<https://github.com/spf13/viper>) for parsing its command-line arguments. All `cobra` projects follow the same development pattern. You use the `cobra` utility for creating the commands, and then you make any changes to the generated Go source code files in order to implement the desired functionality.

Depending on the complexity of your utility, you might need to make lots of changes to the files you've created. Although `cobra` saves you lots of time,

you still have to write the code that implements the sought-after functionality for each command.

Because we're using Go modules, we don't have to download the `cobra` package because this is the job of `go mod init` and `go mod tidy`, which isn't the case for the `cobra` binary utility. We should download the `cobra` utility using `go install github.com/spf13/cobra/cobra@latest` – we can then execute it as `~/go/bin/cobra` and start building powerful command line tools.



The screenshot (page 93) shows the implementation of the time command. Lines 16, 17 and 18 describe the command and its help message. The implementation of the command begins from line 21. The client request to the server is sent in line 31 and the server response is saved in the `resp` variable that has the `http.Response` data type. Type `go doc http.Response` to learn more about the `http.Response` structure. Finally, examine the HTTP status code of the response (line 37). If the response variable isn't empty and the status code isn't `http.StatusNotFound` (HTTP Status Code 404), then everything was okay. We read the body of the response using `io.ReadAll(resp.Body)` and display it on-screen.

Insert command

The insert command requires a JSON record as input. There are two ways to obtain that input: first, as a JSON record; and second, by reading the individual fields and constructing the JSON record internally. Because the `User` structure has only two fields, the latter way is easier, so we'll follow this method. The most important part of the implementation is as follows:

```
req, err := http.NewRequest(http.MethodPost,
"http://"+SERVER+PORT+endpoint, buf)
```

Although we're still using `http.NewRequest()` to construct our request, this time we also send the contents of a buffer, which is the last parameter of `http.NewRequest()`, which holds the payload of the request. In this case, the payload is the JSON record.

The screenshot (below) shows the implementation of the insert command. Line 36 tells the server that we're going to send data in JSON format. This is a compulsory statement because the server needs to know what kind of data to expect. In line 23 we create the JSON record that'll be sent to the server without checking whether the value of the `Username` field is empty or not – we leave that task for the RESTful server. There's no right or wrong way to make checks. However, it's good practice to avoid making unnecessary server requests because these requests create extra network traffic and server load. Finally, in line 48 we check the HTTP status code of the server response – if that status code is `http.StatusOK` then we know the request was processed.

Deleting things

The delete command should read the username variable and put its value into the URL that's going to be visited. This takes place in the following code:

```
19 Long: 'The insert command adds new users to the server DB
20 and requires a JSON record as input.'
21 Run: func(cmd *cobra.Command, args []string) {
22     endpoint := "/insert"
23     user := User{Username: username, Password: password}
24     buf := new(bytes.Buffer)
25     err := user.ToJSON(buf)
26     if err != nil {
27         fmt.Println("JSON:", err)
28         return
29     }
30
31     req, err := http.NewRequest(http.MethodPost, "http://"+SERVER+PORT+endpoint, buf)
32     if err != nil {
33         fmt.Println("Insert - Error in req: ", err)
34         return
35     }
36     req.Header.Set("Content-Type", "application/json")
37
38     c := &http.Client{
39         Timeout: 15 * time.Second,
40     }
41
42     resp, err := c.Do(req)
43     if err != nil {
44         fmt.Println("Do:", err)
45     }
46 }
```

This screenshot show the implementation of the insert command as found in `./cmd/insert.go`. This is the only command that sends a JSON record to the RESTful server.

```
16 var deleteCmd = &cobra.Command{
17     Use: "delete",
18     Short: "delete deletes users given a username",
19     Long: "The delete command deletes all users, given a username.",
20     Run: func(cmd *cobra.Command, args []string) {
21         endpoint := "/delete"
22
23         URL := "http://" + SERVER + PORT + endpoint + "/" + username
24         req, err := http.NewRequest(http.MethodDelete, URL, nil)
25         if err != nil {
26             fmt.Println("Delete - Error in req: ", err)
27             return
28         }
29         req.Header.Set("Content-Type", "application/json")
30
31         c := &http.Client{
32             Timeout: 15 * time.Second,
```

This screenshot shows the implementation of the delete command as found in `./cmd/delete.go`. The command sends no data and receives no data from the server.

```
URL := "http://" + SERVER + PORT + endpoint + "/" +
username
```

```
req, err := http.NewRequest(http.MethodDelete, URL,
nil)
```

So, we first construct the URL that we're going to visit, which includes the username that's going to be deleted, and then we create the request. Because we don't send any data to the RESTful service, the last parameter of `http.NewRequest()` is `nil`.

The screenshot (above) shows the implementation of the delete command as found in `./cmd/delete.go`. Because it occurred with the `insert` command, we don't check whether the value of `username` is empty or not – we leave that task for the server. In line 29 we define that we'll be using the JSON format – this is an unnecessary statement that can be removed because there's no data exchange between the client and the server. However, it doesn't break the code.

Listing things

Finally, let's explain the implementation of the list command. The most important code in `./cmd/list.go` is the following:

```
var users = []User{}
SliceFromJSON(&users, resp.Body)
```

So, why is this code important? The main reason is that the list command receives an array of records from the server, even if that slice is empty or contains just a single record. Therefore, the code should be able to convert that array of JSON records into a slice of structures all at once. As a result, we first create an empty slice of `User` structures and then we call `SliceFromJSON()`, which is a function that decodes a serialised slice with JSON records into a slice of structures and is defined in `./cmd/root.go`. This is the preferred way of transferring and decoding multiple JSON records in a RESTful service.

The screenshot (facing page, top) shows the Go code of `./cmd/list.go`. Apart from the part that has to do with the array of JSON records, most of the code has already been seen in the other commands.

Last, in order to make its output more pleasant, the list command pretty prints the JSON records on screen using the `PrettyPrint()` function, which is located in `./cmd/root.go` and is implemented as follows:

```
func PrettyJSON(data interface{}) (string, error) {
    buffer := new(bytes.Buffer)
    encoder := json.NewEncoder(buffer)
```

```

15 var listCmd = &cobra.Command{
16     Use:   "list",
17     Short: "Lists all data in the RESTful server.",
18     Long:  "This command gets all data from the RESTful server.",
19     Run: func(cmd *cobra.Command, args []string) {
20         endpoint := "/list"
21
22         req, err := http.NewRequest(http.MethodGet, "http://"+SERVER+PORT+endpoint, nil)
23         if err != nil {
24             fmt.Println("List - Error in req: ", err)
25             return
26         }
27         req.Header.Set("Content-Type", "application/json")
28
29         c := &http.Client{
30             Timeout: 15 * time.Second,
31         }
32
33         resp, err := c.Do(req)
34         if err != nil {
35             fmt.Println("List Do:", err)

```

This shows the implementation of the list command – this command might receive multiple JSON records from the server as an array.

```
encoder.SetIndent(empty, tab)
```

```
err := encoder.Encode(data)
```

```
if err != nil {
```

```
    return empty, err
```

```
}
```

```
return buffer.String(), nil
```

All the work is done by `encoder.SetIndent()`, which enables us to apply indent to format the output. The `encoder.SetIndent()` function requires two parameters: the first parameter is what's going to be added at the beginning of each new line and the second parameter is what's going to follow the second parameter. So, in our case, we have no line prefix (empty) whereas each element is indented with a tab. As you're going to see, the end result is a very pleasant output. Inside `./cmd/list.go` there's a `fmt.Println()` call that just prints the generated output.

REST in action

Now that the utility implementation is clear, let's use the utility and interact with the RESTful server. Keep in mind that `cobra` automatically creates a help screen with all available commands and global flags.

The screenshot (*right*) shows the utility in action when interacting with the RESTful server. The first command inserts a user named "Mihalis" in the server database; the second command inserts a user named "mtsouk" in the server database. The third command shows the server contents and `pretty` prints the available records. The next command deletes user "Mihalis" whereas the next command obtains the time from the server. The final command, `go run main.go`, prints the help screen because we haven't given any command to the utility.

For all previous interactions, the server, which was executed as a Docker image as described in part one of this series, has generated the following log data:

```
2021/08/24 11:33:52 Serving: /insert from think:1234
```

```
2021/08/24 11:33:52 {Mihalis passWOrd}
```

```
2021/08/24 11:34:00 Serving: /insert from think:1234
```

```
2021/08/24 11:34:00 {mtsouk passWOrd}
```

```
2021/08/24 11:34:05 Serving: /list from think:1234
```

```
2021/08/24 11:34:15 Serving: /delete/Mihalis from think:1234
```

» DOCUMENTING REST APIS

REST APIs are good but without proper and accurate documentation they're difficult to use. `Swagger` is here to help! Put simply, `Swagger` is a representation of your REST API. `Swagger` reads the appropriate code annotations and creates the OpenAPI file. To be able to document a REST API using `Swagger`, you basically have two choices. First, writing the OpenAPI Specification file on your own (manually), or adding annotations in the source code that help `Swagger` generate the OpenAPI Specification file for you (automatically). The extra content for creating the documentation for the REST API is included in the Go source files of the server as Go comments. The `Swagger` utility reads these comments and generates the documentation. However, all comments should follow certain rules and comply with the supported grammar and conventions.

You can find more about `Swagger` at <https://swagger.io> and about `Go Swagger 2.0` at <https://goswagger.io>. Additionally, you can view OpenAPI files online at <https://editor.swagger.io>. Unfortunately, discussing `Swagger` in detail is beyond the scope of this tutorial.

```
vars: map[username:Mihalis]
```

```
2021/08/24 11:34:23 Serving: /time from think:1234
```

Logging data can be very handy for debugging and troubleshooting server processes.

Finally, if the specified server can't be found, the utility will print the following type of error message:

```
$ go run main.go list -s server
```

```
List Do: Get "http://server:1234/list": dial tcp: lookup server: no such host
```

Go is good at developing web services and clients for RESTful services and this month's tutorial as well as last month's instalment gave you a handy introduction to the topic. If you want to make the most from these two tutorials, you should begin creating your own RESTful servers and clients.

Go resources

The official Go web sites are <https://golang.org> and <https://go.dev>. If you're feeling old skool and prefer to learn from books, then we recommend reading *The Go Programming Language* by Alan A. Donovan and Brian W. Kernighan as well as *Mastering Go, 3rd edition* by this author (*a shocking plug!-ED*). **LXF**

```

+ restclient go run main.go insert -s think -u "Mihalis" -p "passWOrd"
User Mihalis added.
+ restclient go run main.go insert -s think -u "mtsouk" -p "passWOrd"
User mtsouk added.
+ restclient go run main.go list -s think
[
  {
    "user": "Linux",
    "password": "Format"
  },
  {
    "user": "Mihalis",
    "password": "passWOrd"
  },
  {
    "user": "mtsouk",
    "password": "passWOrd"
  }
]
+ restclient go run main.go delete -s think -u "Mihalis"
User with ID Mihalis deleted.
+ restclient go run main.go time -s think
The current time is: Tue, 24 Aug 2021 11:34:23 UTC
+ restclient go run main.go
This is a simple REST API client for Linux Format.

```

This screenshot shows the `restclient` utility in action. In this case the RESTful server is located at a Linux machine named `think` that uses the default TCP port (1234).

» **MAKE SURE WE NEVER REST AND** Subscribe now at <http://bit.ly/LinuxFormat>

On the disc



Discover the highlights from this month's packed DVD!

» START HERE

USING THE LXF DVD

Using Linux for the first time can be very confusing. It'll most likely be unlike anything that you've operated before, especially if you're used to Microsoft Windows or Apple macOS.

Generally our DVDs are designed to be run directly, which is to say that when you first power on your PC (or Mac, see below) it should 'boot' from the DVD – so before Windows or macOS even starts to load – with Linux running directly from the DVD. This trick is known as a Live Disc. It enables you to try out the various versions of Linux without having to install or change anything on your PC. Just remove the DVD, restart your PC and it'll be exactly as you left it.

While many systems will boot from a DVD when it finds one, many will not. See below for the standard process for enabling booting from a DVD on various desktops and laptop PCs.

The alternative option is to locate the ISO file on the DVD and write this to your own USB thumb drive and attempt to run that. We recommend using *Etcher* from <https://balena.io/etcher> that's available for Windows, macOS and Linux. Good luck!

BOOT THE DISC

Many PCs should boot automatically if they're turned on with a disc in the drive. If not, many offer an early Boot Menu accessed by tapping a key while powering up from cold: F9 (HP), F12 (Dell, Lenovo), F8 (Amibios) or F11 (Award BIOS). Alternatively, use the BIOS/UEFI to adjust the boot order to start with the optical drive. Again, this is accessed by tapping a key during power up, usually Del but sometimes F1 or F2.

Some new UEFI PCs require access via Windows: holding Shift select its Restart option. If you're still having problems using the DVD then visit www.linuxformat.com/dvdsupport

Mac owners: Hold the C key while powering on your system to boot from the disc.

LEMUR POWERED

MIN SPECS: 4GB RAM, 25GB HD SPACE

Ubuntu 21.10

64-bit

We have to wait until April 2022 for the Ubuntu 22.04 LTS release. In the meantime we've got the last of the interim releases, which besides giving some hints as to what to expect in the LTS, is a damn fine OS in and of itself. The release was just too late to feature the Gnome 41 desktop (which made it into the delayed Fedora), though some apps have been co-opted from the new release. Gnome 40 is nothing to be sniffed at, either: it's got dynamic workspaces and laptop users will love the support for three- and four-fingered touchpad gestures.

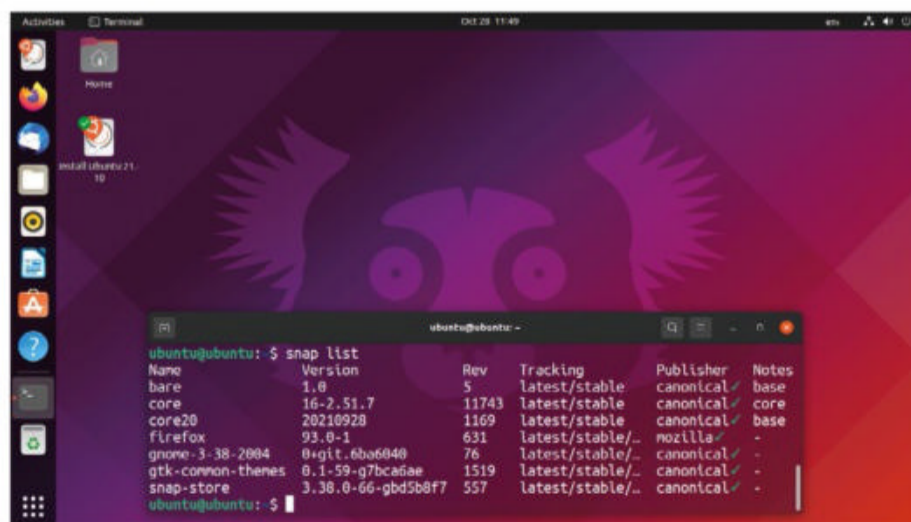
Snaps, Canonical's homespun application packaging and containment system, gains yet more prominence in this edition. As is the norm, a few Gnome programs and the associated runtime are bundled in Snap form. But the big change is that now *Firefox* is too, and that the *Firefox* Snap comes straight from Mozilla. Our quest to get hardware video decoding working with this Snap (in the cover feature) may not have been a success. But that was just teething problems that in all likelihood will be remedied by the time you read this. Having the default browser inherit the security benefits that come with Snaps, and

having it shipped straight from the source, will be a boon for everyone.

We often complain about distros' installation media getting larger and larger, so imagine our shock and awe when we saw that this one was a whopping 3.1GB. Just as well Devuan remains small enough that Ubuntu could have some company on the DVD. There's not any one reason why the ISO has grown so large this release. Kernel images get bigger, more and more firmware gets bundled, the Nvidia driver has to fit on there, and those base Snaps really do rack up the megabytes. In an age of exabytes and CPUs with dozens of cores though, it's silly to call this bloat and spend a paragraph whining about it.

We know some readers will only be interested in the desktop release, but there's an awful lot to like as regards to your servers, cloud instances, Edge and IoT devices. In the enterprise sphere, Canonical offer their own Ubuntu-powered take on Kubernetes. Recent work here with Nvidia has enabled GPU passthrough for containers. Even if you're not an enterprise type, you can still install Canonical's MicroK8s and have a fully fledged, multi-node Kubernetes stack in a few commands.

With Ubuntu 21.10, you can now fire up an NGC (Nvidia GPU-Optimized Container) on Nvidia hardware (such as its Jetson range) and get started with machine learning. See www.brighttalk.com/webcast/6793/481263 for a demonstration.



Snap, a-ahh! Saviour of the application-packaging universe. Unless you're using Mint, Pop!_OS or other snap-shunning distros.

» IMPORTANT NOTICE!

DEFECTIVE DISCS: For basic help on running the disc or in the unlikely event of your *Linux Format* coverdisc being in any way defective, please visit our support site at www.linuxformat.com/dvdsupport. Unfortunately, we're unable to offer advice on using the applications, your hardware or the operating system itself.

FREE AS IN MYTHICAL CREATURES

MIN SPECS: 1GB RAM, 10GB DISK, LESS WITHOUT A GUI



» AND MORE!

THE LXF LIBRARY

- **Advanced Bash Scripting Guide**
Go further with shell scripting.
- **Bash Guide for Beginners**
Get to grips with the basics of Bash scripting.
- **Bourne Shell Scripting**
First steps in shell scripting.
- **The Cathedral and the Bazaar**
Eric S Raymond's classic text explains the advantages of open development.
- **The Debian Book**
Essential guide for sysadmins.
- **Dive Into Python**
Everything you need to know.
- **Introduction to Linux**
A handy guide full of pointers for new Linux users.
- **Linux Dictionary**
The A-Z of everything to do with Linux.
- **Linux Kernel in a Nutshell**
An introduction to the kernel written by master hacker Greg Kroah-Hartman.
- **The Linux System Administrator's Guide**
Take control of your system.
- **Tools Summary**
Overview of GNU tools.
- **GNU Emacs Manual**
Six hundred pages of essential information!
- **Producing Open Source Software**
Everything you need to know.
- **Programming from the Ground Up**
Take your first steps.

Devuan 4.0 "Chimaera" 64-bit

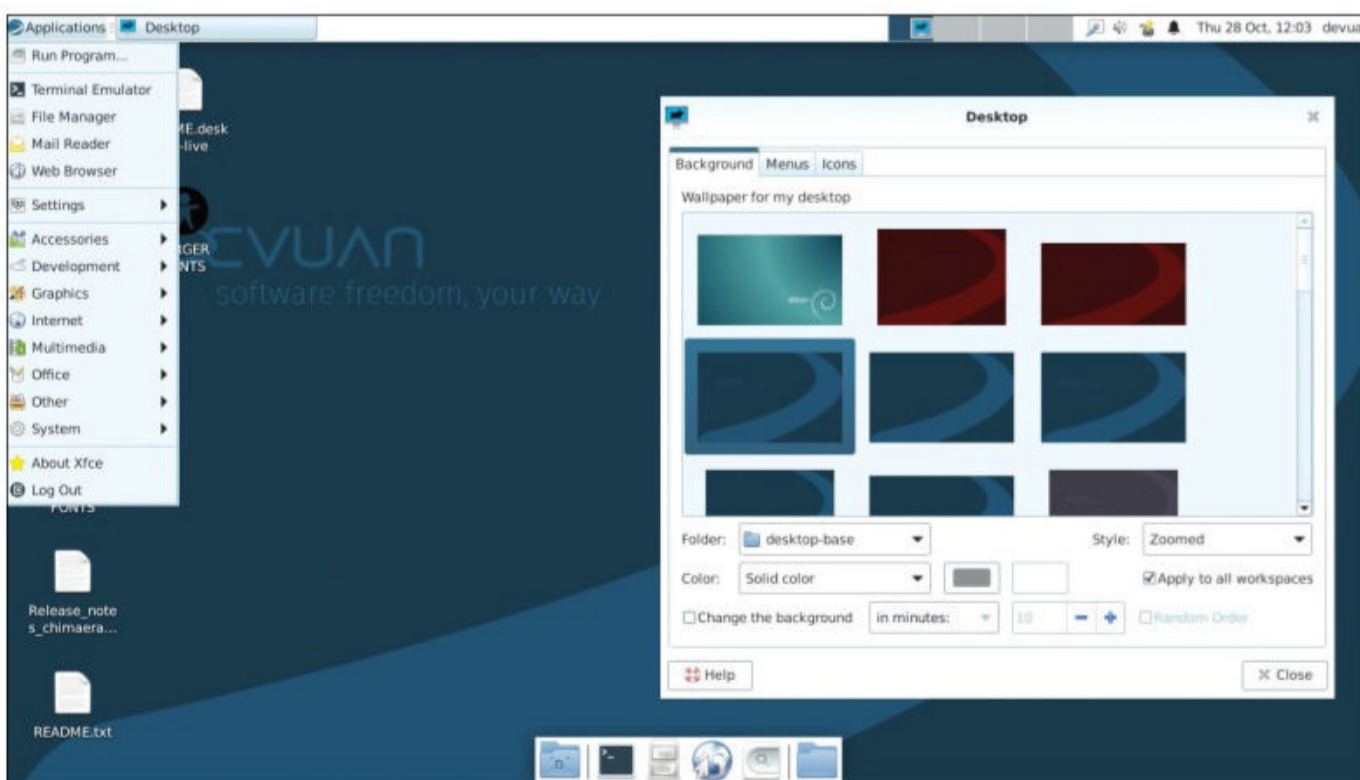
We haven't given much attention to Devuan lately, which is remiss of us because it's surely the world's premier Systemd-free distro. Devuan offers a choice of the old (sysvinit), the new (runit) and the in-between (OpenRC). Besides that, it's based on Debian 11.1, features a 5.10 series kernel and can now run "virtually all" desktop environments in the Debian repositories. This is good because we still see unenlightened tech journalists bashing GNOME for being dependent on Systemd, when GNOME has for many years been fully functional through elogind, the Systemd-free login daemon.

Pulseaudio is also optional in this release, and if you want to use *Pipewire* for audio then it'll probably be much more difficult to set that up in Devuan than in Ubuntu (which only involves installing a couple of packages) or Fedora (where it's working out of the box). Devuan inherits all the new accessibility features from the Debian installer too. So visually impaired people can use a Braille display or speech synthesis. This support extends into the installation, and one reason for not wanting *Pulseaudio* is that it prevents speech synthesis being available to both GUI and console applications. The *Orca* screen reader from GNOME performs the oration, and if you don't need it (or any speech

synthesis) then you can opt out at install time.

You also have a choice of login/display managers. The lightweight *Slim* is still the default, but the slimline *lightdm* is also available. New in this release are *sddm* and *gdm3*, the ideal gatekeepers for KDE Plasma and GNOME, respectively. Like Debian, Devuan is still doing a 32-bit (i386) release, as well as supporting PowerPC (ppc64el) and three different flavours of ARM architecture. There are also mini, netinstall-style ISOs as well as unofficial Docker images available.

Oh, and we'd love to have put the live Xfce desktop environment on the DVD, but once again the ginormous Ubuntu ISO made this impossible, so we only have the server edition. You can add the Xfce desktop with `sudo apt install task-xfce-desktop`. There are instructions for installing only the core Xfce bits and pieces at www.devuan.org/os/documentation/dev1fanboy/en/minimal-xfce-install.html, if you prefer a more minimal desktop experience. If you're already running Beowulf, the previous Devuan release, then it's simple to upgrade. And likewise if you're running Debian Bullseye then it's easy to migrate with a few edits to your Apt sources file. Again, you'll find comprehensive instructions on the website.



Devuan's bundled background offerings don't amount to much. We were hoping for an actual picture of a chimaera.

» NEW TO LINUX? START HERE...

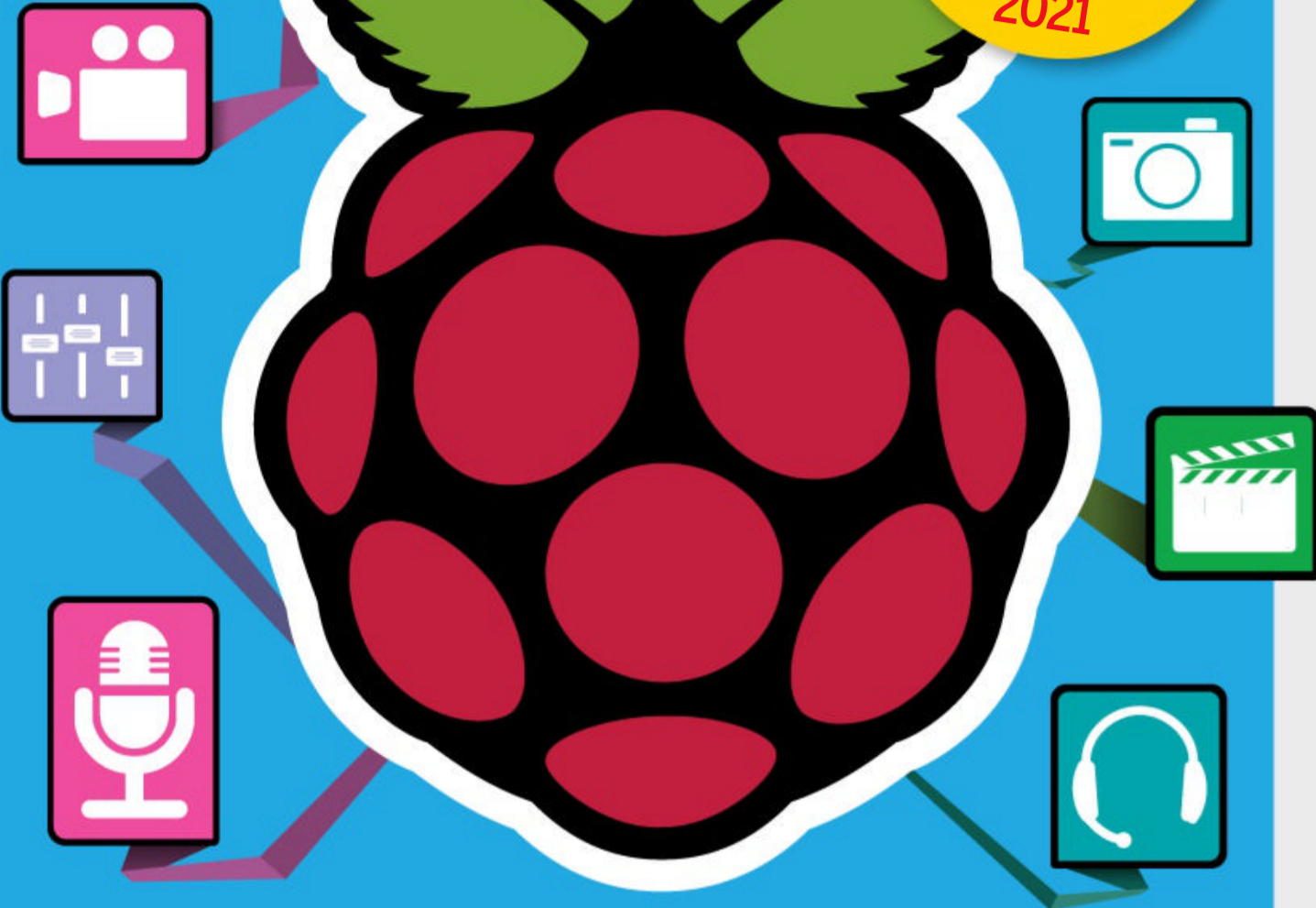
Never used a Linux before? Here are some handy resources:

- Read our quick-install guide <http://bit.ly/LXFinstall>
- Looking for an answer? <https://askubuntu.com>
- Want to delve more deeply? <https://linuxjourney.com>



NEXT MONTH

LXF284
will be on sale
Tuesday
14 December
2021



RASPBERRY Pi MEDIA CENTRE

Combine the mighty Pi, LibreElec 10 and loads of cables to build an awesome streaming centre!

Super Linux!

Discover how Linux went from the research lab to powering every top 500 supercomputer in the world today!

Next-gen encoding

Open source is finally driving next-gen high-efficiency video codecs, so discover the best ones to use.

Switch to Btrfs

We guide you through the transition to the next-gen storage system know as Btrfs and how to avoid losing all your data!

Start a recording studio!

We kick off a two-part guide to creating your very own Linux-powered recording studio. It's easier than you think!

Contents of future issues subject to change – we might not be able to afford our gas bill.

LINUX

The #1 open source mag

FORMAT

Future Publishing Limited,
Quay House, The Ambury, Bath, BA11UA
Email linuxformat@futurenet.com

EDITORIAL

Editor Neil Mohr
Chess Master Jonni Bidwell
Art editor Efrain Hernandez-Mendoza
Operations editor Cliff "Still on holiday" Hope
Production Ed Ricketts, Katharine Davies
Group editor in chief Graham Barlow
Group art director Jo Gulliver
Editorial contributors
Mats Tage Axelsson, Michael Bedford, Neil Bothwick, Sean Conway, Matthew Hanson, Mathew Holder, Jon Masters, Nick Peers, Les Pounder, David Rutland, Andrew Smith, Shashank Sharma, Mihalis Tsoukalos, Alexander Tolstoy

Cover illustration magictorch.com
Ubuntu is a trademark of Canonical Limited. We are not endorsed by or affiliated with Canonical Limited or the Ubuntu project. Linux Fedora and the Infinity design logo are trademarks of Red Hat, Inc. Raspberry Pi is a trademark of the Raspberry Pi Foundation. Tux credit: Larry Ewing (lewing@isc.tamu.edu) and *The GIMP*.

ADVERTISING

Commercial sales director Clare Dove
clare.dove@futurenet.com
Senior advertising manager Lara Jaggon
lara.jaggon@futurenet.com
Head of commercial – Technology Dave Randall
dave.randall@futurenet.com
Account director Andrew Tilbury
andrew.tilbury@futurenet.com

INTERNATIONAL LICENSING

Linux Format is available for licensing and syndication. To find our more contact us at licensing@futurenet.com or view our content at www.futurecontenthub.com.

Head of Print Licensing

Rachel Shaw

NEW SUBSCRIPTIONS & BACK ISSUES

Web www.magazinesdirect.com
UK 0330 333 1113 **World** +44 (0) 330 333 1113

EXISTING SUBSCRIPTIONS

Web www.mymagazine.co.uk
UK 0330 333 4333 **World** +44 (0) 330 333 4333

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

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. All copyrights and trademarks are recognised and respected. **Disclaimer** All contents © 2021 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 2008885) is registered in England and Wales. Registered office: Quay House, The Ambury, Bath BA11UA. 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: Uh oh, what's the price of petrol doing?

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.

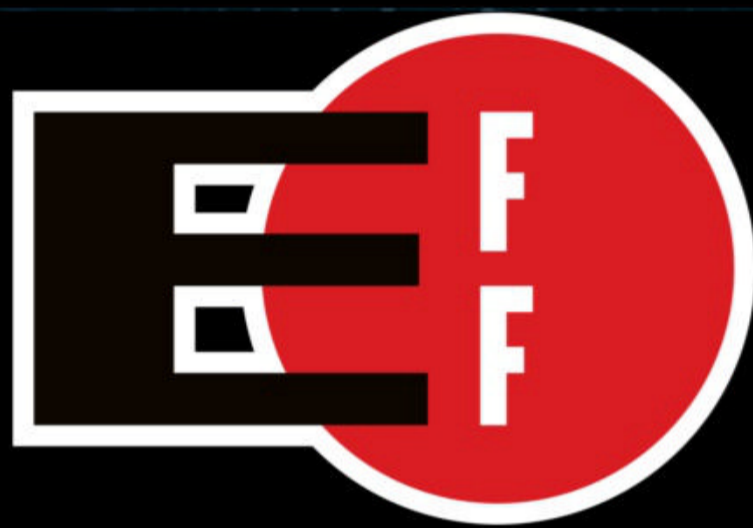


Future is an award-winning international media group and leading digital business. We reach more than 57 million international consumers a month and create world-class content and advertising solutions for passionate consumers online, on tablet & smartphone and in print.



Future plc is a public company quoted on the London Stock Exchange (symbol: FUTR)
www.futureplc.com
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



THE UNIVERSITY
of EDINBURGH



Swansea
University
Prifysgol
Abertawe



QUEEN'S
UNIVERSITY
BELFAST

SIGN UP

