LINUX FORMAT

The **#1** open source mag

HACKER SPECIAL
Pop on your white hat & start pentesting today

NEW RELEASE!

UPGRADE TO UBUNTU!

Learn how to use the most popular Linux distro ever! It's free, slick, secure and offers years of support updates

HAPPY BIRTHDAY

20 years of Ubuntu!

PLUS: HOW TO

» Build a Pi Pico oscilloscope
» Write Commodore 64 games in C
» Use AI to create better office documents

FEAR Y2K 2.0!
The year 2038 cometh and we're all doomed!

CLASSIC DEMOS
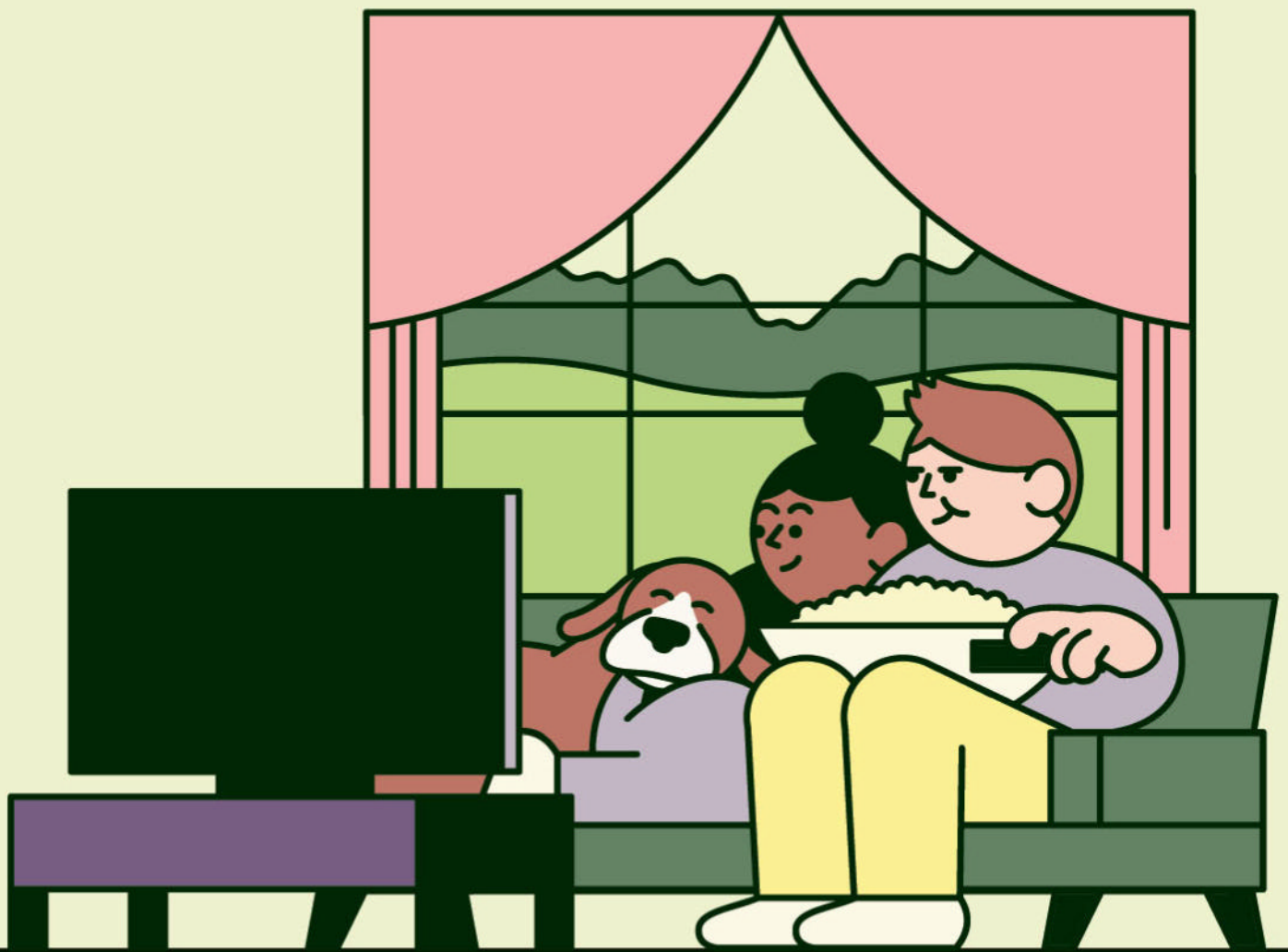Code classic Amiga demos in simple C

SWEET SOUNDS
Make beautiful music with enhanced USB

LINUX FORMAT

Corporate Patron
Free Software Foundation

» MEET THE TEAM

Ubuntu has turned 20 and we're celebrating everyone's favourite orange distro. So, what's your oldest Ubuntu memory and is it your daily driver?

Jonni Bidwell
I didn't see what all the fuss was about when Warty was released. It seemed like Knoppix had already figured out hardware detection a year earlier. Maybe I was more contrarian back then, because I quite liked the brown/orange stylings. And yes, I do do dat daily drive.

Neil Bothwick
I remember including this brown distro on an **LXFDVD** and thinking it will never catch on. I also recall the furore over the decision to use *sudo* and disable root login. People seemed upset that they had to type four more characters to run something inadvisable.

Nick Peers
My earliest memory of Ubuntu has to be 14.04 LTS, which became my first permanent desktop Linux installation. And I've actually stuck with it ever since – on the desktop, at least (I'm currently dabbling with Debian Server in order to benefit from its superior compatibility with *Podman)*.

Les Pounder
I've been an avid Ubuntu supporter since 5.04, but it was in 2010 that I ran my first installfest at Oggcamp* in Liverpool. For an entire weekend, Aijaz, Stuart and myself installed Ubuntu on laptops for the delegates. Challenging and fun times were had by all.

Mayank Sharma
I can't believe it's been two decades since the spaceman turned Debian into a distro everyone could use, and mailed a disc of their first release to anyone who asked. The desktop distro isn't my daily driver, but I can't escape running into its Server variant on the public cloud.

*https://bit.ly/lxf315ogg

Orange hero

You either die a hero or live long enough to see yourself become the villain. It may be a film cliché but often there's more truth told in jest than many might want to admit. If you read certain parts of the internet, you'd think that's what happened with Ubuntu. Snaps have destroyed users' lives, *Systemd* has infected mainstream distros, Ubuntu Pro is pushed on unsuspecting users, no 32-bit support, that awful desktop design, toxic corporate culture, the list goes on.

But there's another cliché: you can't please all of the people. Because Ubuntu, by many metrics, is still easily the most popular Linux distro in the world and that's largely because it just works – the thing that grabbed the world's attention in 2004 with the original Warty Warthog release is what keeps people using Ubuntu today in 2024.

Twenty years on and we're celebrating the latest 24.04 LTS release and taking a look back at how Ubuntu exploded on to the Linux world stage, how it developed over the years and what the major controversies were, as there's been a few…

One change that we don't think will be controversial is Canonical extending (paid) Ubuntu support to 12 years. That's going to help people feel even more comfortable installing Linux on their systems, ensuring a consistent and prolonged user experience without the unnecessary changes and forced upgrades that other OSes can demand. My home server has been ticking away with Ubuntu since 14.04 LTS and I expect many more years of Ubuntu enjoyment to come!

Neil

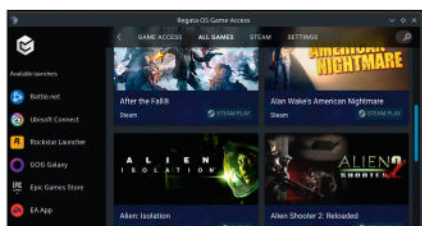**Neil Mohr** Editor
neil.mohr@futurenet.com

Subscribe & save!
On digital and print: see page 16

Contents

REVIEWS

Les Pounder finds a fun if flawed tiny laptop that caters to RISC-V fans and runs Debian 12 and some of the tools you need.

Let's hope that in the course of this distro review, **Neil Mohr** doesn't make any gags he's going to regata. Damn it!

Nate Drake peers underneath the hood to find Arch-based Cachy is flashy – but is it also a solid distro to use?

Nate Drake goes comfortably numb at the vast number of improvements introduced in Ubuntu 24.04 Noble Numbat.
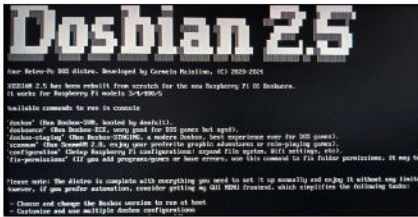
UPGRADE TO UBUNTU 24.04 LTS

The latest Ubuntu is here. **Jonni Bidwell** has come out of hibernation just to celebrate and (in between cake and tea) tell you all about it. See page 30!

CREDIT: Magictorch
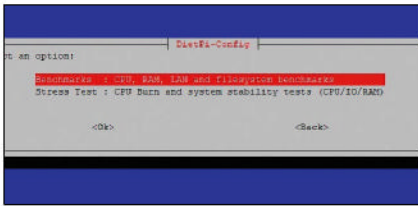
ROUNDUP
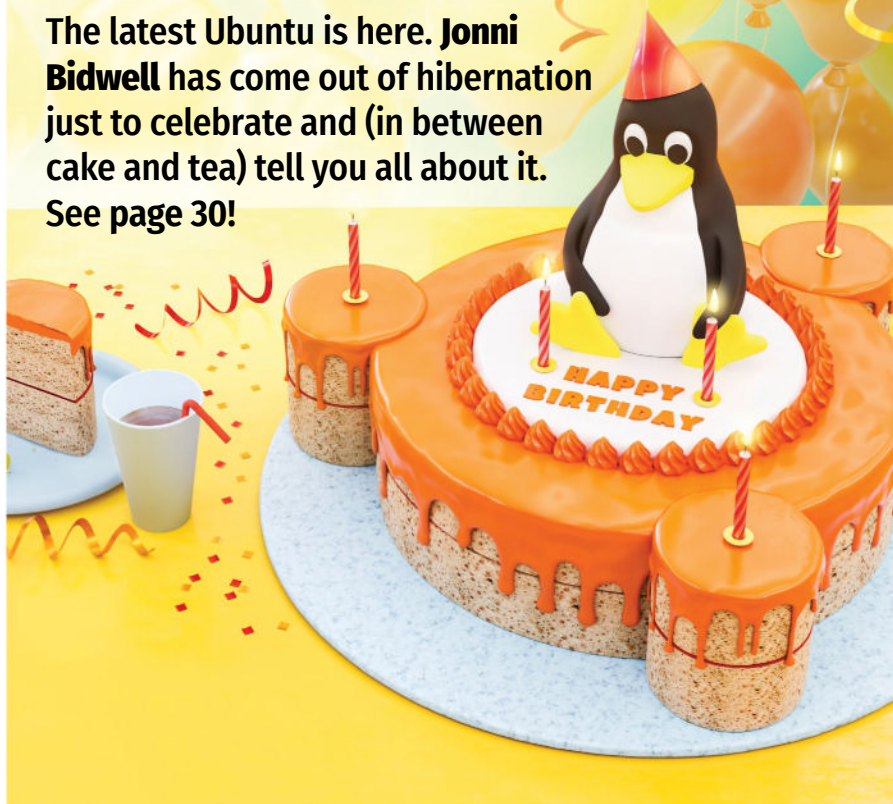
Donning his best white hat, the always ethical **Michael Reed** examines five Linux pentesting distros while hacking into your smart fridge to check your milk levels.

IN DEPTH

A decade on from celebrating a decade of Ubuntu, **Neil Mohr** wonders where all the time has gone as he takes a trip through the history of the venerable distro.

CREDIT: Canonical

www.linuxformat.com

CONTENTS

Pi USER

CODING ACADEMY

REGULARS AT A GLANCE

TUTORIALS

ADMINISTERIA

Newsdesk

THIS ISSUE: Linux back door found ≫ Rust-based Nvidia driver in pipeline ≫ Snap apps snap ≫ FlyOS flies on to phones

HACKING

Microsoft engineer spots Linux back door

In late March, Microsoft software engineer Andres Freund noticed a tiny delay when logging in via SSH, and uncovered a mysterious back door.

In late March 2024, Microsoft software engineer Andres Freund was flying home to San Francisco from his native Germany. He'd been doing some micro benchmarking and saw that his system's *sshd* processes were using an unusual amount of CPU resources. This in turn was generating a number of errors in *Valgrind*.

Andres explored further, finding this was caused by error messages centred around liblzma, one of the major components of *XZ Utils* along with *xz* itself. The source code for both of these are publicly available via GitHub, as are the associated binaries. *XZ Utils* can be found in almost every version of Linux, given that it provides lossless data compression.

Freund initially believed that this issue only affected his own OS (Debian Sid). After more careful investigation, he discovered that in fact the upstream *xz* repository and the *xz* tarballs had been back-doored. This back door affected versions 5.6.0 and 5.6.1 of *XZ Utils* and worked via additions to the configure script in the TAR files, making it very difficult to detect.

The implications of this would have been that any server using this version of *XZ Utils* and running SSH could potentially be accessed using a special private key. In the event, the vulnerability was patched within hours but had it remained undetected, it would almost certainly have found its way into popular upcoming server distros, such as Debian 13 and Ubuntu 24.04.

This has led to much online speculation as to how a major vulnerability could remain undetected for so long. After all, taking a collaborative open source approach to software development is supposed to prevent precisely this kind of scenario.

From examining the source code, it's clear the back door into *XZ Utils* was inserted by a co-maintainer using the name Jia Tan. Tan apparently created a GitHub account in November 2021 under the username JiaT75, and the following year submitted a patch via the mailing list. At this point, other accounts surfaced using the name Jigar Kumar and Denis Ens, making feature requests and encouraging creator Lasse Collin to accept the patch and take Jia Tan on as a co-maintainer.

Three days afterwards, Jia Tan started making regular commits to *xz*, effectively taking control of the project.

The huge level of sophistication involved over a number of years in implementing the back door has led to speculation by the likes of Costin Raiu of Kaspersky that a state actor was behind them. In the meantime, Andres Freund has been given the dubious title of the Silverback Gorilla of Nerds.

You can view latest updates on the back door at **https://tukaani.org/xz-backdoor/**.
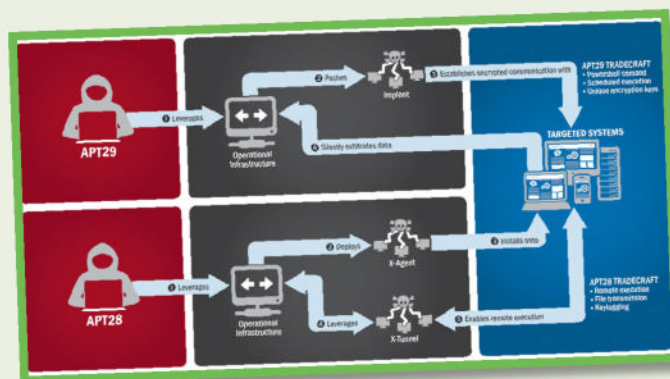
The pattern of this attack superficially resembles that of APT29, aka Cozy Bear, who has links to Russian foreign intelligence.

Nvidia Rusty RedHat driver

Red Hat has started work on a Rust-based GSP-only driver for Nvidia GPUs.

On 20th March, Red Hat display driver software engineer Danilo Krummrich announced the creation of the Nova project, a Rust-based driver for Nvidia GPUs.

This is no doubt welcome news to users of the open source graphics driver Nouveau, given that last year its chief maintainer at Red Hat resigned from the project.

In the developer mailing list, Krummrich announced that Nova is intended to succeed Nouveau, with the aim to create a simplified, more modern driver: "Nouveau's historic architecture, especially around nvif/nvkm, is rather complicated and inflexible, and requires major rework to solve certain problems (such as locking hierarchy in VMM/MMU code for VM_BIND, currently solved with a workaround), and second, with a GSP-only driver, there is no need to maintain compatibility with pre-GSP code."

Danilo also appreciated the virtual mountain to climb when it comes to Rust development, given the missing C binding abstractions for integral kernel infrastructure, such as device and driver abstractions. Currently, the project aims to start with a basic device/driver, DRM and PCI

In theory, the Nova driver could provide better Linux compatibility with Nvidia RTX 3000 graphics cards.

abstractions, along with a Nova stub driver to make use of them.

The benefits of a Rust-based driver include increased memory safety and potentially better performance, given features such as the language's minimal runtime overhead.

Online reaction has been positive, but some Redditors point out that Nova isn't the only drop-in replacement for Nouveau. Collabora has been working on its NVK open source Vulkan graphics driver since October 2022 and has recently promoted it to the stable channel, so it will be added to the Mesa 24.1 graphics stack.

View the current repository at **https://gitlab.freedesktop.org/drm/nova/**.

Oh, Snap! Crypto malware ahoy!

More malware is discovered in Ubuntu Snap Store.

In February, former Canonical developer Alan Pope blogged about the fact that one user had lost nine bitcoins (currently worth around £500,000) due to a fake version of the *Exodus* wallet they'd installed via the Ubuntu Snap Store.

The Snap was malware masquerading as a legitimate crypto wallet in order to persuade users to hand over their private 'seed'.

The app in question was removed a few days later and Ubuntu CEO Mark Shuttleworth issued a statement, saying: "Our goal should be that anybody using Snaps from the official Snap Store on any distro should be safer than if they were getting that software from other hosting platforms." He went on to admit the scope of the problem, whereby cybercriminals

were using social engineering to impersonate popular developers, but stopped short of an all-out ban on crypto apps in the Snap Store.

Pope published another blog post in mid-March, stating that very little had changed since then, as he claimed to have discovered that 10 more "scam bitcoin wallet apps" had been published in the Snap Store that day.

It seems Canonical is now taking the issue more seriously. In late March, Canonical representative Holly Hall announced via the Snapcraft forums that all new Snap registrations will be manually reviewed.

In early April, Flathub also modified app download pages to make it clearer if a program is unverified – in other words, uploaded by someone other than the original developer.

SQL AT FIFTY

Dave Stokes is a technology evangelist at Percona.

Structured Query Language (SQL) is 50 years old. It is the only programming language from the '70s that is still around and – unlike COBOL and Fortran, which are confined to limited niche roles – still thriving and expanding. In 2023, the IEEE declared SQL the most popular programming language.

But why is SQL still around? It is certainly not easy to learn its syntax. Every database vendor has a quirky implementation that does not port directly to another one. Forgetting a WHERE clause can easily wipe out an entire table in an instant.

But when combined with the relational model of databases, it maps very well to many business practices. The ability to perform transactions is the core of most processes today. SQL was the first programming language to return multiple rows per single request. And by segregating customer information in one table and manufacturing data in another, you can easily compartmentalise data.

There are attempts to dethrone SQL. NoSQL databases promised to replace SQL but couldn't make it happen. Occasionally natural language processing advocates try to take over, but end up with lugubrious kludges that get ignored. For all its clunkiness, SQL just keeps on going.

OPINION

ACCESS ALL AREAS

Italo Vignoli
is a founder of LibreOffice and
the Document Foundation.

"*LibreOffice 24.2* introduced several new accessibility features, thanks to the presence of a developer dedicated to this specific area.

Accessibility refers to the ability of computer systems to deliver services and information usable, without discrimination, by those who, due to disabilities, require assistive technologies or special configurations.

LibreOffice has millions of users, so has to offer the possibility of using it barrier-free to everyone, especially those who have barriers in the form of disabilities.

Moreover, accessibility features often also benefit users who do not have problems, as they simply make the software better.

For example, the description of images – which is often omitted out of simple negligence, but is a fundamental element for the document to be accessible (as it is read by the screen reader to those who are visually impaired or even blind) – can contribute to a better understanding of the images themselves for all other users.

LibreOffice 24.8, which will be announced in August, will offer more features in the area of accessibility. In fact, this is a road that has been taken and will bring significant improvements to *LibreOffice* over the next few years."

DISTRO

FlyOS offers Linux subsystem on Android

The FlyOS project claims to transform a phone into a pocket-sized computer.

Once installed, FlyOS offers the potential to run Linux desktop apps like Wine via the device's web browser.

DigitalPlat, FlyOS developer, proudly announces on its main page: "Imagine coding on your phone, connecting a keyboard and mouse, and effortlessly accessing your favourite tools like *LibreOffice, VSCode, GIMP, Jupyter* and more. It's a complete Linux system that fits in your pocket!"

Although many of these activities can be achieved on Android via *Termux* and *Linux Deploy,* FlyOS is optimised for mobile devices. Users use the device's web browser to interface with the containerised version of Linux. The developers draw comparisons between FlyOS and WSL (Windows Subsystem for Linux), which uses virtualisation to run a Linux kernel inside a lightweight utility virtual machine.

In order to install on Android, users must first unlock the bootloader and have rooted their device – via *Magisk,* for example. They can then install the FlyOS software on their PC (currently this is for Windows only) in order to connect the Android device and write the software to it.

Discover more about the project at **https://github.com/EdwardLab/flyos/**.

CENSORSHIP

Turkey bans Distrowatch

Website is banned, supposedly for offering dangerous software.

Since 2001, Distrowatch has helped Linux users make informed decisions by posting news and reviews of various versions of the OS. In late March, the National Cyber Incident Response Center (USOM) in Turkey decided to block access to the site for local users, though it's not entirely clear why.

USOM claimed that Distrowatch's IP hosts "harmful software" and it's a source of malware.

Given the internet censorship in Turkey, a more cynical explanation is the government doesn't want its citizens to discover privacy-orientated distros to protect themselves online.
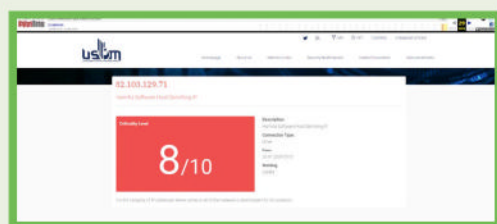
The original USOM announcement has been taken down and is only accessible via the Wayback Machine.

LICENSING

Redis licence drama

In-memory database vendor switches to more restrictive dual licence.

On 20th March 2024, Redis announced that as of version 7.4, the software will be dual-licensed under the Redis Source Available Licence (RSALv2) and Server Side Public Licence (SSPLv1).

The Open Source Initiative characterises the Server Side Public Licence as a fauxpen licence. Organisations that use the licensed code for SaaS are required to disclose the entire source code of the service to users, rather than solely their modified version of the software.

An open source fork has now been created, which uses the LGPL. See **https://redict.io/**.
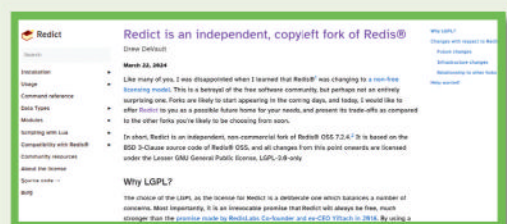
Redict is a fork of Redis 7.2.4, which used the BSD 3-clause licence. Future updates are licensed under LGPL 3.0.

Distro watch

What's behind the free software sofa?

ROSA 12.5

Rosa Linux is developed by Russian company LLC NTC IT ROSA. The first stable releases came out in 2010 as a fork of Mandriva, but Rosa is now developed independently. The latest version (12.5) uses kernel version 6.6 and offers a number of different desktop environments (Gnome, KDE Plasma, LXQt, Xfce) for x64, i686 and AArch64 architectures. There's also a console-only version for experienced users. Read more at **https://rosa.ru/**.

Rosa is a Russian Linux distro, originally forked from Mandriva.

AV LINUX MX EDITION 23.2

This is a bug-fix release for the Debian 12-based multimedia distribution that's aimed at content creators. It's built on the high-performance Liquorix Kernel and uses the Enlightenment desktop, with this edition based on MX Linux. With version 23 shifting to Debian 12 and moving to PipeWire, this point release is mopping up some showstopping bugs that had been introduced with the new base. You can read the full details at **www.bandshed.net/2024/04/05/av-linux-mx-edition-23-2-iso-update/**.

AV Linux got a solid review back in LXF306.

4MLINUX 45.0

4MLinux is presumably so-named because it's designed for four main purposes: Maintenance, Miniserver, Multimedia and 'Mystery' (games). The latest version includes new print drivers. *FFmpeg* has improved support for video scaling via the zimg library. Stereo audio is improved via libbs2b. The distro's *DOSBox* package now includes arcade game *CHAMP Kong*. Other games are playable via Mesa 23.3.0 and *Wine 9.2*. OpenSSL is now version 3. Learn more at **https://4mlinux.com/**.
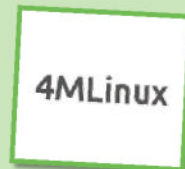
4MLinux is a mini distro, focused on four core functions.

OPENBSD 7.5

OpenBSD enjoys the distinction of being one of the oldest continually developed operating systems, having first been released in 1996. The project emphasises portability and proactive security. The latest release marks the 56th since the project began and incorporates new features across the board. These include bt and btrace support for the binary modulo operator (%); drm has also been updated to Linux 6.6.19. For a full list of all of the OS's new features, visit: **www.openbsd.org/75.html**.

OpenBSD is a Unix-like OS designed with security in mind.

PEROPESIS 2.1

As we noted in our review of Peropesis in **LXF304**, its developers describe the project as "incomplete" but subject to constant improvement. In the case of the latest version, said improvements include new packages for *GNU Autotools*, such as *Automake, Autoconf, M4, Libtool, pkg-config, Diffutils,* Perl, zstd *and lzip.* A number of packages have also been updated, including OpenSSL, which is now version 3.1. Discover more at **https://peropesis.org**.
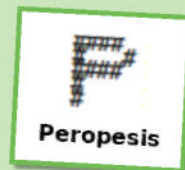
Peropesis is an ultra-lightweight command-line version of Linux.

NVK IS GO!

Faith Ekstrand
is an engineering fellow at Collabora.

❝ NVK, the open source Vulkan driver for Nvidia hardware in Mesa, is ready for prime time. The merge request has landed, getting rid of the non-conformant implementation warnings and signalling to distros that it's now time to start shipping NVK to users. I can't speak on behalf of distros, but NVK will be part of Mesa 24.1 and you should expect to see it in either the spring or autumn release of your favourite Linux distro.

In October, I announced that NVK had reached Vulkan 1.0 conformance on Turing hardware. As of now, NVK is a conformant Vulkan 1.3 implementation on Turing (RTX 2000 and GTX 1600 series), Ampere (RTX 3000), and Ada (RTX 4000) GPUs. Not only have we jumped forward three Vulkan versions, but the new test runs were done with the GSP firmware enabled and includes Ampere and Ada GPUs. Also, unlike the initial 1.0 run, there are no hacks. Each test we passed in those conformance test runs also passes on upstream Mesa.

Performance is still a work in progress and continues to improve regularly. A lot of titles are running at 60fps or better on recent GPUs. With others, we're seeing bottlenecks that we have yet to triage. If you want to know if your favourite game performs well, the best way is to just try. ❞

OPINION
THE X TO Z OF EXPLOIT

Jon Masters is a kernel hacker who's been involved with Linux for over 22 years.

I wanted to take a moment to discuss the broader implications of the XZ attacks involving maintainer burnout. The XZ library had a single maintainer for years, one who (by their own admission) was experiencing a tough moment, causing them to feel real burnout. As a result, patches were languishing and releases were not as regularly finished.

All of a sudden, an apparent well-wisher named Jia Tan appeared and began making useful contributions. Eventually, other 'people' put pressure on the maintainer to accept them as a co-maintainer. So it was that this 'actor' was able to exploit a maintainer to gain control over a project.

What's lost in all of the discussion about the fallout and consequences is that this is just one of the bullying tactics applied to gain leverage over an open source project. This is far from a new phenomenon. Look around for a few minutes and you'll find similar bullying tactics elsewhere, used to force maintainers to hand over control. If we are to move beyond the XZ moment, we should also consider reforming other aspects of how our communities operate, including how we handle pester efforts by folks to force maintainers to do things they would not otherwise do.

Kernel Watch

Jon Masters keeps up with all the latest happenings in the Linux kernel, so you don't have to.

Linus Torvalds announced the fourth RC (Release Candidate) for what will become Linux 6.9 in another few weeks. In his announcement, he noted that there was "Nothing particularly unusual going on this week – some new hardware mitigations may stand out, but after a decade of this I can't really call it 'unusual' anymore, can I?" This was in relation to yet another speculative execution vulnerability that had been mitigated, called Native Branch History Injection.

HISTORY REPEATING ITSELF

"Some new hardware mitigations may stand out, but after a decade of this I can't really call it 'unusual' any more, can I?"

The original BHI was disclosed by researchers at VUSec back in March 2022. It exploits a CPU's branch predictor logic (in particular, the history of previous branches, known as the BHB or Branch History Buffer) by effectively creating a phantom history that can be used to mistrain future predictions and cause the CPU to perform (measurable and attacker-controlled) accesses that can be used to reconstruct sensitive data. At the time it was felt that finding the necessary 'gadgets' (exploitable code) within the kernel was sufficiently hard that the mitigation was to disable unprivileged eBPF, so that an attacker couldn't simply load the code they needed into the kernel for an attack. In the latest attack, a tool was created to find alternative gadgets within existing kernel code that are exploitable. The fix is to bring x86 in line with other architectures by zeroing the BHB when crossing privilege levels.

Fallout is fun

On 29th March, Andres Freund, a developer working at Microsoft, announced that he had – through sheer luck and aptitude for troubleshooting performance issues on his systems – found a back door subtly hidden in the upstream XZ Utils, a popular compression library used on many FLOSS systems. The actual attack is now well documented, relying on a subtle compromise to build artefacts not directly part of the source, but widely used to build distro packages. The new result being that the attacker(s) are able to compromise SSH connections on *Systemd*-based distros, which is pretty much most of them by now. Things could have been much, much worse, were the back door not detected as early as it was.

You can, of course, learn much more about this elsewhere within these pages and online. But what might have been missed by others reporting more broadly on this incident was that the attacker(s) were also quite possibly in the process of preparing to attack the Linux kernel, too. Linux uses XZ compression for various parts of the kernel image (depending on the build configuration). This requires a separate implementation of XZ within the kernel itself, used to decompress these artefacts at runtime. At the time of the XZ attack, a recent patch had been posted for review (but not yet actually merged), documenting the attacker as an authorised maintainer of the kernel code.

No doubt they intended, eventually, to leverage this for more nefarious purposes, targeting the Linux kernel as well. LXF

>> DEVELOPMENTS

There's an ongoing effort to refactor the kernel's PCIe subsystem and move various code out of the older devres (device resource) framework in the aim of consistency. Philipp Stanner posted the third version of his work.

In an unusual turn of events, Linux kernel developers from Apple posted a patch to the kernel mailing list (which almost elicited a cheer from some quarters), intended to facilitate the use of x86-TSO memory model emulation on Apple Silicon machines running Linux guests, so that they can run Rosetta.

Answers

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

Neil Bothwick can answer anything once he's fully charged.

Q Cinnamon challenge

I was originally using the Cinnamon version of Linux Mint 21.3. I heard good things about MATE, so I did a new installation to try out MATE. After several months, I decided that I would like to go back to Cinnamon. I am not sure how to switch back without reinstalling Mint again – and because this Mint PC is my daily driver, I don't want to do a new reinstall again as it's very time-consuming to configure things how I want again.
Anthony Harris

A

The only significant difference between the Cinnamon and MATE releases is the default set of desktop packages installed on top of the base Linux Mint software. If you want to switch to Cinnamon, you only need to install the **mint-meta-cinnamon** package. Meta packages like this are a way of installing multiple packages as a set; **mint-meta-cinnamon** is not much more than an empty package that depends on the various packages used by the Cinnamon edition. Once you have installed this, log out of the MATE desktop and on the login screen, you'll see an option to select your preferred desktop, just to the right of your login name. This means you now have both Cinnamon and MATE available. The system remembers your last choice and makes it the default next time, which is particularly useful if you use auto-login. Having MATE there as well should make no difference to your computer apart from a slight consumption of disk space. Adding Cinnamon on top of MATE used around 200MB here; MATE is lighter so expect to save less than that by removing it. In the past, we have seen some duplicate menu entries when installing multiple desktops but that does not seem to be an issue now.

If you really want to get rid of MATE, you can uninstall the **mint-meta-mate** package. This in itself does not free up any space as it is only a meta package, but you should then be able to remove any packages that were only dependencies of this one with:

```
$ sudo apt autoremove
```

We recommend taking a backup before any autoremove operation, just in case it removes more than you wanted. As for having to reconfigure things after a reinstall, you may wish to consider using a separate partition for **/home** when you come to install the next version. Per-user configuration settings are saved in your **home** directory, so keeping this separate from the operating system means your settings, and other data, are not touched when you install a new OS.

Adding another desktop environment is easy, with no need to remove the existing one. Then you have a choice of which to use when you log in.

Q Not so new

I currently have a home server running *Openmediavault*. I use *TVHeadend* to record TV shows to a folder that is automatically scraped into *Kodi* by *Jellyfin*. I have an issue, however. When the episode is new, the folder is named with **New:** in the title and this confuses *Jellyfin*. Is there any way to automatically remove the phrase "New:" from the beginning of any folder in my **TV Shows** folder?

Also, because of the way that *Jellyfin* removes recordings, I often end up with empty folders. Is there any way to also automatically remove any empty folders from my **TV Series** folder as well?
Bradley North

A

Both of these can be resolved with *Find*. Taking the easiest task first, removing empty directories, you can use something like this:

```
$ find /folder -mindepth 1 -type d -empty -exec rmdir "{}" +
```

The first argument is the folder to search. The **-mindepth** option means *Find* only looks for results below that directory, otherwise the first directory returned is the search directory. **Type** restricts the search to directories and **-empty** does exactly what you would expect. Finally, **-exec** runs the command given, with **{}** replaced by a list of the search results, while **+** terminates the **-exec** command. The quotes around **{}** are needed in case any of the names include a space. You would need to run this multiple times as deleting a folder may make its parent empty, which will be picked up on the next run. You can have empty parents of deleted folders removed at the same time by using **-p** with **rmdir**. This has the side effect of producing lots of output as deletion of non-empty folders is attempted, but you can silence that with **--ignore-fail-on-non-empty**.

By now, you should realise you can also find folders with "New" in their name with:

```
$ find /folder -type d -name "New: *"
```

The **-name** option uses shell-style wildcards, so this matches any folder
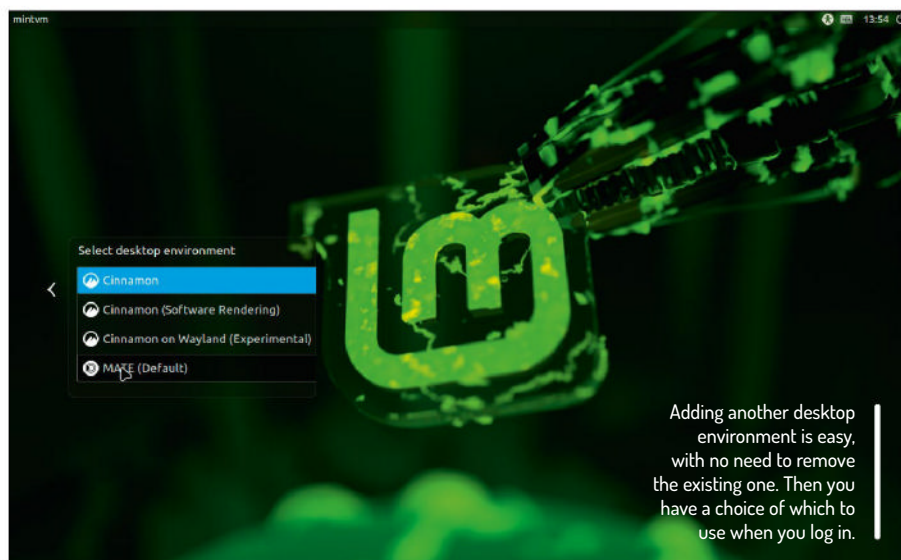
whose name starts with "New: ". If you prefer regular expressions to shell globbing, use **-regex** instead of **name**. Now all you need is a command to rename the directory, which is called *Rename.* The basic syntax is

```
$ rename ABC DEF file1 file2 ...
```

This replaces the first occurrence of ABC with DEF in each file's name. This is what you want, as you've said the folder names start with "New: ". There are options to *Rename* to alter this, detailed in the man page, but we don't need them here. So, to replace "New: " with the empty string, your *Find* command becomes:

```
$ find /folder -type d -name "New: *"
-exec rename "New: " "" "{}" +
```

There is one thing to be aware of: there are several *Rename* commands available for Linux, each with their own syntax. We are using the 'standard' *Rename* command that comes with the **util-linux** package that is installed on all distros. If may be safer to use the full path, **/usr/bin/ rename**, especially if you run this scripted automatically and unattended.

Backup choices

**Q** I'm seeking a backup utility that will copy from my download directory to a removable drive that does not attempt to sync the directories. I don't want anything deleted from the target directory, and I don't want anything added to the source, and I don't need anything already present in the target to be rewritten just because it's in the source.
Luke Glover

A It sounds like *Rsync* is exactly what you want. Yes, it syncs but it has numerous options to have it do exactly what you want. The default behaviour is pretty close to what you require; it only copies files from your source to the destination if either they do not exist on
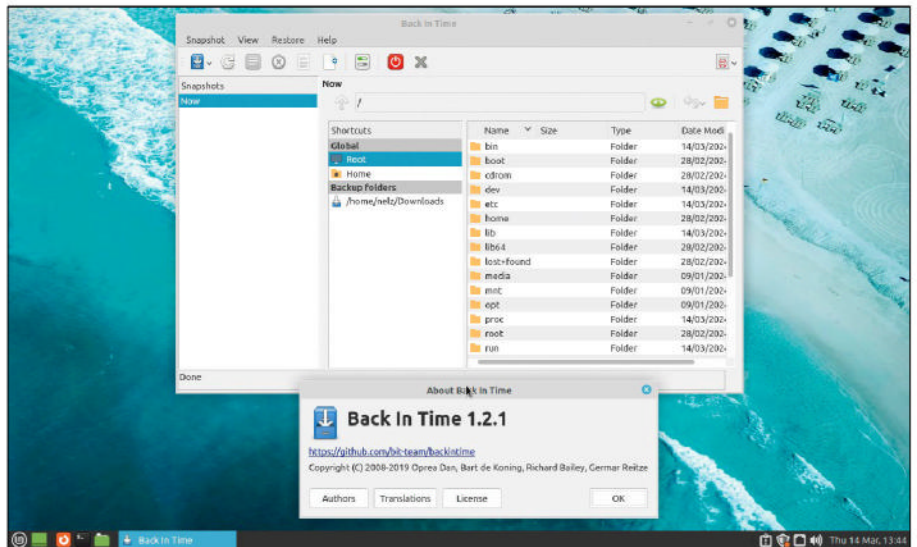
the destination or if the source file has changed. It does not delete files on either side unless told to do so. For example, to copy new and changed files, preserving permissions and timestamps, use:

```
$ rsync -p -t Downloads/ /media/
removable/
```

Rsync is fussy about path specifications, so make sure you include the trailing slash on both paths, otherwise it doesn't do what you expect – the *Rsync* man page explains this further. It's not clear what you mean about not syncing the directories. If you mean you do not want subdirectories copied, this command is what you want. If you are referring to the deletion of removed files but you do want subdirectories copied, then add the **-r** option. Or you can use **-a** (**--archive**), which covers **-p**, **-t** and **-r**. There are many other options detailed in the man page but two that are useful when first using *Rsync* are **-v**, which lists the actions it is performing, and **-n** (or **--dry-run**). The latter doesn't do anything, but in combination with **-v** shows

you what it would have done. You should use this to check your command options.

This is a terminal command, but you could add it to a desktop launcher. If you prefer a graphical program, there's a number of options that provide a clickable face to *Rsync*, such as *Grsync* (**http://www.opbyte.it/grsync/**) and *Back In Time* (**https://github.com/bit-team/backintime**).

Deep search

Q I need to search through a large number of emails for some specific information. All of the emails are in one folder in *Thunderbird,* but many of them contain the important information inside PDF files attached to the emails. While I can search the email bodies, how can I run a search across all the attachments?
Kieran Davey

A *Thunderbird* doesn't provide this ability natively, and saving all the emails in the folder to perform an external search will not help as *Thunderbird* does

---

⟩⟩ A QUICK REFERENCE TO... DISK SPACE

No matter how large hard disks grow, we always end up running out of space. One reason for disk space wastage is duplicate files. You may have saved the same video in different locations, for instance, possibly with different names, or downloaded the photos from your camera or phone more than once. Finding duplicate files that may be in various directories on your hard disk

and possibly with different names could free up a lot of wasted space. That is where *Fdupes* (**https://github.com/ adrianlopezroche/fdupes**) can help.

This is a command-line program that scans the given directories and reports any duplicate files it finds. To be duplicates, the files must have identical contents, but not necessarily the same names, timestamps or

owners. A basic invocation would be:

```
$ fdupes -r ~
```

Note that *Fdupes* considers only directories, not filesystems, so if you run it on **/** it happily descends into virtual filesystems like **/proc** as well as any network mounts you have. The default output is to list each group of duplicates one file per line with a blank line between groups. The **-1** (**--sameline**)

option shows all files in a group on the same line, while **-f** (**--omitfirst**) omits the first match, producing a list that is a starting point for cleaning up your hard drive. The **-d** (**--delete**) option prompts you for which file(s) in a group to keep, while **-n** (**--noempty**) excludes empty files from the search. There are usually plenty of these and they are not duplicates, each having their own purpose.
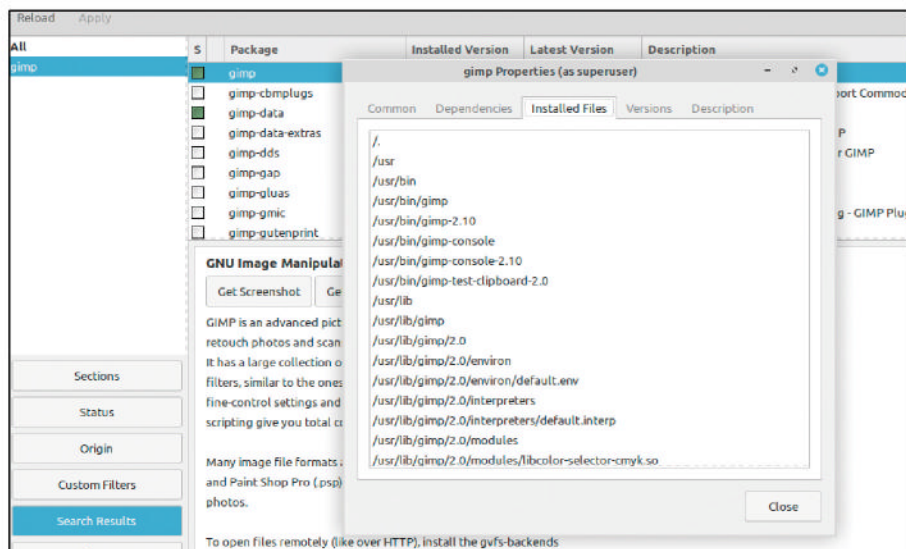
Package managers do not install all files for a package in a single location, but do keep track of where they go.

not save the attachments in the EML files it exports, although this may be possible with an extension. The simplest option is to grab the entire folder from the mail server with *Offlineimap,* which should be installable from your distro's repositories. While in your package manager, install *Mpack* and *PDFgrep.*

There are some example config files in **/usr/share/doc/offlineimap/examples**; copy **offlineimap.conf.minimal** to **~/.offlineimaprc**. Edit it to add your mail server and username. To download only one folder, add this to the remote section:

```
folderfilter = lambda folder: folder.
endswith('foldername')
```

You may also need to add `ssl = no` if you get complaints about missing certificates. To download the folder, run:

```
$ offlineimap
```

Now you can extract all the attachments with *Munpack* from the **mpack** package. Change to the directory containing the mails you just downloaded and run `$ munpack *`.

This extracts all attachments from all the mails. Now you can search all the PDF attachments with:

```
$ pdfgrep 'search term' *.pdf
```

The command-line options for *PDFgrep* are similar to the standard *Grep* command. It's a little cumbersome, having to do it in three stages, but you could put the commands into a shell script if this is something you will need to do regularly.

Scattered files

Q I have installed *GIMP 2.10.30* on my Ubuntu 22.04 desktop, using the standard repos. Can you tell me the folder to which the files were installed? I need to check whether a number of Python plugins have been installed.
Jodie Page

A Linux packages are not normally installed in individual folders. Generally, executable files are stored in **/usr/bin**, libraries in **/usr/lib**, data files for servers in **/var/lib**, man pages in **/usr/share/man** and most other files in a programs's directory in **/usr/share**. If that seems a mess, bear in mind that files, particularly libraries and executables, can be used by more than one program, so placing them in a standard location makes them accessible to all. So, how do we keep track of all these files? We don't, the package manager does. For Debian/Ubuntu, this is *Apt*. The *Apt* command itself doesn't list package contents, but its low-level cousin, *dpkg,* does.

```
$ dpkg -L gimp | less
```

Here we are piping its output through the pager **less** so that we can read it all. Press Space to scroll down to read the full output, or you can press `/` followed by a search term to seek what you are looking for. You can also pipe the output through *Grep* to search for a file:

```
$ dpkg -L gimp | grep Restore
```

If the files you want are not installed, they are probably contained in a different package. To find that you need a large database of all the packages available and what they contain. You can find this at **https://packages.ubuntu.com**. Enter the file you want and hit the search button to find the package that will install it.

You can also see the list of files in an installed package in the *Synaptic* graphical package manager – you may need to install this. Right-click on a package and select Properties to get a window that includes a tab listing the installed files.

Group bug

Q I tried *QEMU/*KVM six months ago and I can't make it work in my system,

so I removed it, but these errors are persistent – how do I remove them?

```
systemd-udevd[288]: /usr/lib/udev/
rules.d/50-udev-default.rules:102
Unknown group 'kvm', ignoring
systemd-tmpfiles[284]: /usr/lib/
tmpfiles.d/static-nodes-permissions.
conf:17: Failed to resolve group 'kvm'.
```
Matthew Bond

A This is odd, because we looked at a vanilla Linux Mint 21.3 installation and these udev rules were already present, despite having no *QEMU* or other virtualisation software installed. The KVM group also existed, so the rules did not generate any errors. After installing and uninstalling *QEMU*, the rules and group still existed. It appears that at some point, when removing all trace of *QEMU,* the KVM group was removed, either through the package manager or by your action.

The simple solution is to recreate the KVM group with the same ID it had in the default installation, with this command:

```
$ sudo groupadd --system --gid 109
```

It is not essential to use the same group ID, but 109 is what Linux Mint used by default on our test system. The messages are not harmful – they are warnings rather than errors – but recreating the group should mean they annoy you no more.

This is why taking a *Timeshift* backup before making changes to your system is beneficial. Then you would have had the option of rolling back, although that would only have been helpful if you decided to remove *QEMU* soon after installing it, with no other system changes in between. **LXF**

Mailserver

Funny fonts

I've just installed Linux Mint. It was very easy and it's great, but the fonts are awful, as they have always been on Linux. The first thing I did was increase the font size to 15 in the system preferences, and turn on hinting to full. This made fonts look fuzzy, but at least they aren't jagged.

But on some websites, the fonts are still jagged. I tried installing *Fontconfig* and editing **fonts.conf** to force no bitmap fonts anywhere and to always use Liberation Sans for everything, but this had no effect. What's a good way to get better-looking fonts?
Tegan Sharmila

Neil says…

It's not something I hear people complain about these days – font rendering used to be a real mess – but browsers can also cause their own issues. So, it does depend if you mean system-wide or just a browser problem. Open the System Settings in Linux Mint and in the Fonts/Appearance section, set Font Rendering to Default or Restore Default Fonts. We recommend rebooting to ensure everything is applied.

If it's a case of missing default fonts, you can try installing the generic Microsoft web fonts with:

```
$ sudo apt-get install msttcorefonts
```

If you're still having rendering issues, you might want to try flushing the font cache with:

```
$ fc-cache -fv
```

Cute Krita

I'm looking into switching from Windows to Linux. I have been considering this for a while but the main

Fonts in 2024 certainly look better than back in Ubuntu 4.10 days, that's for sure.
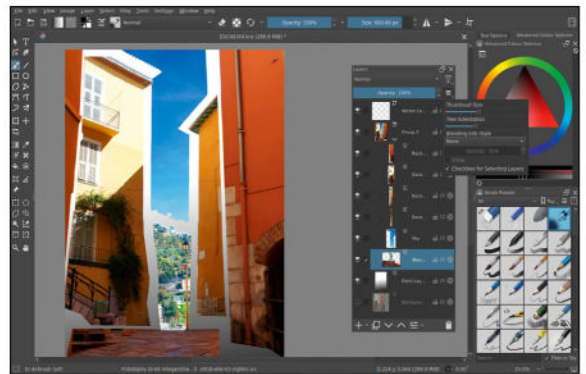
Krita is built for tablet and stylus work – ideal for digital artists.

art package I use isn't available for Linux. However, with changes to Windows 12, I'm being given more of a push to jump and was looking at *GIMP* and *Krita*.

What I'm stuck on, however, is finding a good Linux distro to try. I would like something that provides as novice-friendly an experience out of the box as I can get. However, it needs to play well with Huion art tablets and art software like *Krita* and *GIMP*, and also work for gaming.

My three main contenders are Ubuntu, Kubuntu and Pop!_OS, all primarily because of how well established and widely used they are, so I would have access to a lot of support. If anyone has any insight, personal experiences of how these distributions perform with art and gaming, or better suggestions, I would like to hear them.
Trillian Hudson

Neil says…

I hate to say it, but you can run *GIMP* and *Krita* on Windows, so ideally you should try them out before doing anything drastic. Having said that, it's great that you want to shift over to Linux and, honestly, any of the three distros you've mentioned should offer a decent experience. The Huion drivers are natively supported by Linux, so that shouldn't be an issue, and *Krita* is built from scratch for tablet use by digital artists *(make sure you read our tutorial on p.60)*.

As for gaming, these days Steam provides pretty decent support on Linux. You can check any games you play at **www.protondb.com**. The main problem for Linux gamers are online multiplayer games, where anti-

Helpdex

It's a PowerPC-based system branded as the AmigaOne from A-EON.

cheat largely remains a Windows-only thing, although there are noises about some coming to Linux, thanks to pushes from Valve, but we're not holding our breath.

Amiga mad!

Please could you review the latest *AmiKit* for the Raspberry Pi 5 and general Linux? And how about looking at Linux running on the Amiga X5000 or A1222 Amiga systems?

Ian Learmonth

Neil says…

We managed to get a look at *AmiKit* in **LXF309**, so you should hunt down that tutorial. I missed the release of the A1222+ in 2023 by the looks of things; let's see what we can do. Let us know what you think. Find out more: **https:// amigang.com/ hardware-amigaone-a1222/**

Yes, we did indeed make our virtual machines run a lot faster.

»LETTER OF THE MONTH

Virtually impossible

I'm a very big fan of your magazine and wanted to make a suggestion for a potential article. I recently came across one that named details about virtualisation *(Qemu, VirtualBox,* etc) and containers *(Docker, Podman,* etc), and I had something I have been working on for a long time now and would like to know if this is a possibility for an article: GPU virtualisation and/or partitioning.

I can't say whether this is important to the magazine or not, but knowing how to do that would be very exciting for security experts. Usually on Nvidia, Intel and AMD (my GPU brand of choice) graphics cards, you cannot do this because the drivers lock down the card from doing so, meaning all the graphical and security features are impossible. However, I know *Hyper-V* does it, but (sadly) I haven't come across a way to do it with a good amount of graphics cards on Linux, other than *Libvf.io* (**https://bit.ly/lxf315multi**) with only select GPUs, and *VGPU Unlock* (**https://bit.ly/lxf315lock**), but I suspect I am crossing some legal red tape with the driver.

I have also heard of para-virtualisation like *Easy-GPU-PV* (**https:// bit.ly/lxf315gpupv**) but, once again, that's Windows-only. If anyone knows if this is possible, it would be the journalists or editors of this magazine. I have searched everywhere, asked AI, looked at GitHub repos and more, but only *Hyper-V* is one of the options, and Windows is not the topic here nor my OS of choice. I was just wondering if there is any way to do that and, if any of the journalists specialise in virtualisation, whether they can help me and how to get in contact with them?

I have been looking for two years but nothing has worked. If anyone has an option similar to *Hyper-V* to the point it's like this (**www.youtube.com/watch?v=ZQxEwC6lyco**), that would be amazing and I would appreciate it.

Jon

Neil says…

We have looked at this before, and when I say "we" I mean middle-aged genius Jonni has looked at both para-virtualisation (**LXF244**) and also the more recent Intel GVT-g (**LXF291**) GPU virtualisation. The issue with the latter, which is the much more useful and interesting option, is that it requires more up-to-date hardware for it to work at all, never mind all the correct software configuration, which can be less than straightforward. I think Intel's sixth-generation processors released at the end of 2015 were a minimum, which should be less of an issue now. There's a decent guide on setting up KVM on Ubuntu with GVT-g to run Windows clients here: **https:// bit.ly/lxf315kvm**

Lichee Console 4A

Les Pounder finds a fun, if flawed, tiny laptop that caters to RISC-V fans.

Looking like a ThinkPad that has been shrunk *[a ShrinkPad? – ed],* this little laptop isn't a netbook but a RISC-V development platform. Inside is a RISC-V-powered SoM (system on module) SO-DIMM board connected to a carrier board. The SoM is a LicheePi 4A, similar to earlier Raspberry Pi Compute Modules. The SoM has the RISC-V CPU, RAM and eMMC storage baked in. Accessing the GPIO, HDMI and Ethernet ports requires a carrier board, in this case the laptop's mainboard. Under the hood is a 1.5GHz quad-core RISC-V CPU and our review unit came with 16GB of RAM and 128GB of eMMC storage.

The red pointing stick is clearly aping the TrackPoint found on many ThinkPad machines. The Lichee is small, 180x140x20mm, but it feels tough enough to throw in a bag as we move around the world. There is a USB port on the right side, next to a headphone jack.

The screen is a pleasant 7-inch LCD touchscreen, with a 2MP camera on the left bezel. The screen looks like a Google Nexus 7 2013 has been grafted on to a keyboard, and on boot we can see the boot sequence rotated 90 degrees anticlockwise. When we get to the desktop, though, normal orientation is resumed. The aluminium alloy back is strong but we advise you to open the screen at the centre and not on a corner.

Watch this space

It all feels a little cramped and while the pointer works, it's stiff and no amount of tweaking fixed its slow vertical tracking. The keyboard is a compromise, with just 72 keys and a split spacebar. How many times did we hit the pointer when we wanted the spacebar? Lots.

Our unit came with Debian 12 and after setting up Wi-Fi and connecting to our router, we updated the OS to the latest packages. The 16GB of RAM and 1.5GHz quad-core RISC-V CPU provided a smooth experience.

To write a new OS image to the SoM, we need to set up our machine running Linux and prepare the Lichee for flashing. Problem is, we can't access the internal hardware and the carrier board has a button we need to hold in order to enter boot mode. Pesky soft metal screws prevented us from taking the case off.

Booting in 46.9 seconds isn't great but this isn't a device built for blistering boot speed, rather for RISC-V development on the move. In general use, Lichee runs Debian well. We cold-opened *LibreOffice Writer* in 15.46 seconds and it was ready for work. Opening *Chromium* took 6.2 seconds, fast enough for us.

The 1,280x800 display suggests we can watch at least 720p video, and for locally held files, this is true. We tested a 4K video using the *Parole* video player. It played fine, but this was on the internal screen, not a 4K display. Streaming YouTube videos at 720p is possible, but not 720p 60fps. Those videos just tank.

Connecting an external screen to the mini HDMI port gave us two screens. Initially, this was limited to

We know we've been getting bigger over the years, but this is ridiculous!

768p. After a firmware update, we were greeted with a 1080p external display, but we'd borked our Wi-Fi. The swings and roundabouts of development hardware!

The 128GB eMMC is fast enough, around 262MB/s when testing a sequential read. This is around PCIe Gen 1 speeds and exceeds the 235MB/s recorded for both the Pineberry Pi Hat Drives and Pimoroni's NVMe Base for the Raspberry Pi 5. Write speeds are decent, at 82.8MB/s. For those interested in comparing these speeds against microSD cards, using SDR104 on the Pi 5 we saw 90.5MB/s read and 30.8MB/s write.

The underlying OS is Debian Linux and that means our favourite software packages are just an `apt-get` away. Well, some of them are. We wanted to install the *Arduino IDE,* to see if we could write a simple program for the venerable microcontroller. There was nothing in the repositories, nor was there an installation candidate for RISC-V on the Arduino website. We also hit this issue when trying to install *Sysbench* for our benchmarks. If the software that you need is in the *Apt* repositories for RISC-V machines, great news. If not, you had better learn how to compile from source. **LXF**

VERDICT

DEVELOPER: Sipeed
WEB: https://sipeed.com/licheepi4a
PRICE: £367 (16GB, 128GB eMMC)

FEATURES	8/10	EASE OF USE	5/10
PERFORMANCE	5/10	VALUE	6/10

An impressive and tiny 'ShrinkPad', which captures the look and feel of the ThinkPad, while offering a RISC-V CPU.

» Rating 6/10

Regata OS 24

Let's hope **Neil Mohr** doesn't make any gags he'll regata. Damn it!

IN BRIEF

An OpenSUSE-based distro from Brazil, maintained for almost a decade, this is a slick, well-presented distro with a custom gaming angle that can double as your daily driver.

SPECS

CPU: 2GHz dual-core 64-bit
Mem: 4GB
HDD: 30GB
Builds: x86-64

There are two likely reasons we've not heard much about Regata OS. One: it's from Brazil. Two: it's based on OpenSUSE. In a world that seems to revolve around Debian/Ubuntu-based distros in the English language, stepping away from either is bound to create speed bumps. So, is offering a slick games-first distro enough to move past them?

We first looked at Regata OS 22 in **LXF293**; 18 months on and two versions later, we have Regata OS 24 Arctic Fox, with the big update being KDE Plasma 6 desktop alongside a jump to kernel 6.7 and Mesa 24. Let's tackle those last two, as between them they offer support for the latest gaming graphics hardware. Mesa 24 improves AMD Radeon Vulkan ray-tracing performance and new Vulkan extensions to the Intel ANV driver. This enables the Regata OS Games Access system to use VKD3D-Proton 2.12, which enables supported DirectX 12 games, primarily those built on Unreal Engine 5, to run out of the box. With this, KDE Plasma 6 provides the best Wayland support so far – we'll have more on KDE 6 next issue.

Get down with the downloads

Regata OS is offered in two downloads, one generic and one with the Nvidia display driver, something other distros offer to help smooth dealing with the proprietary Nvidia driver. At 3.4GB, it's clearly a balanced build. The ISO is both a live disc, so you can try it out without installing, and straight install media.

The distro uses the platform-agnostic *Calamares* installer, but it has customised and split it in two. From within the live environment, the installer helps partition the disk and copy the files. The user creation process only shows up after you boot into your new installation.

The whole raison d'être of Regata OS is gaming. At its heart is its own *Game Access* launcher; this ties together the major gaming systems such as GoG, Epic Store, Ubisoft, Origin and Battle.net alongside Steam,
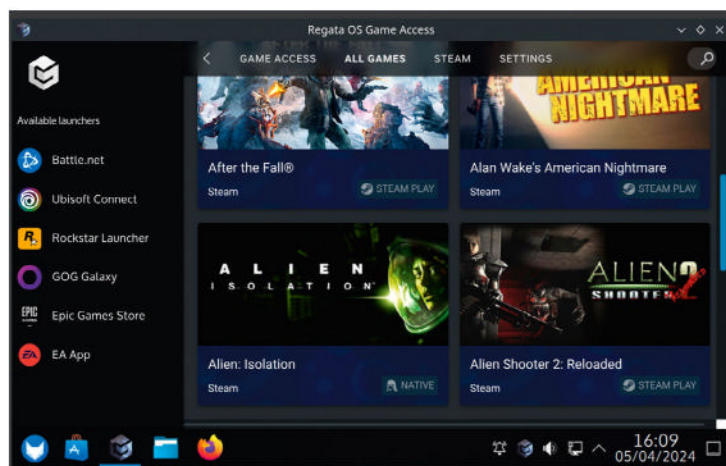
All your games in one place.

though oddly Steam is installed outside of the tool and it's one job left up to you. It's basically a *Lutris*-like system in distro form, but why not just use *Lutris*? Install the store, log in and compatible games are ready to launch. As it's leveraging Steam Play and *Proton*, it can't work magic, so games that use incompatible anti-cheat systems, such as *Fortnite,* still won't work, but in terms of enabling simplified gaming access, it's a win.

What if you want to make it more of a practical daily driver distro? The developers only bundle a spartan selection of default tools. The big names are *Okular* and the newly added *OnlyOffice*, providing office support alongside *Firefox* and *Kate* for an IDE. It does go to explain the smaller ISO size. To move beyond this basic selection, just open up the Regata OS Store.

The excellent support documents and guides are visually appealing, perfect for the novice. These are backed up with a Telegram channel for live support from the community. A little worrying, the OpenSUSE forums aren't impressed by the selection of repos used by Regata OS and it does turn off GPG checks and auto-refresh, alongside certain security repositories. Hopefully, this is something that can be resolved. **LXF**

A slick Settings tool helps complement the default Yast.

VERDICT

DEVELOPER: Marcos Queiroz Gomes
WEB: https://get.regataos.com.br/
LICENCE: Various

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

A slick novice-friendly presentation with helpful guides, and its pre-installed gaming chops should turn a few heads.

» **Rating 7 /10**
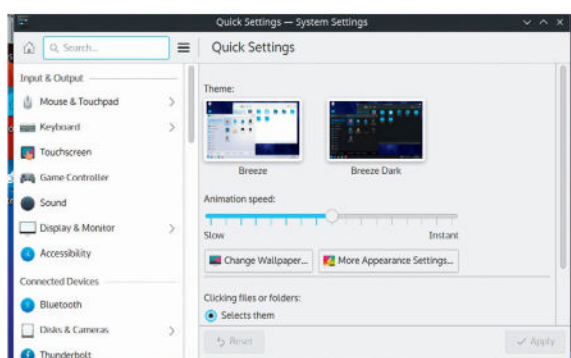
CachyOS 240401

Nate Drake peers underneath the hood to find Arch-based Cachy is flashy – but is it also a solid distro to use?

Stable versions of CachyOS have been released since late 2022. The OS itself is based on Arch Linux but it's been heavily customised.

Some of these changes are centred around speeding up the kernel using the *BORE (Burst-Oriented Response Scheduler)*. The desktop packages are compiled with LTO (Link Time Optimisation), as well as enhancements for x86-64-v3 and x86-64-v4 compatibility.

While talking customisation, of all the Linux distros we've reviewed, CachyOS is one of the most flexible given that the downloadable GUI installer offers virtually every known desktop environment and window manager, including KDE, Gnome, Xfce, CuteFish, i3wm, Wayfire, LXQt, OpenBox, Cinnamon, UKUI, LXDE, MATE, Budgie, Qtile, Hyprland and Sway.

We tried to download the 2.8GB ISO three times from the CachyOS website but it failed. Luckily the site also contains links to download via SourceForge. The live DVD boots via KDE, which now includes Plasma 6.

Away with Wayland

The March 2024 ISO defaults to Wayland sessions, though the latest April release has reverted to X11. The release notes state that this was due to issues with how the *Calamares* installer interacted with Wayland. We also noted that Gnome ISOs are no longer offered for download, though the setup assistant happily loads the desktop environment if you proceed to install.

If you do opt for the installer, you'll also see that CachyOS 240401 offers three different bootloaders for online installation: systemd-boot, Grub(3) and rEFInd. It also provides a new rEFInd partitioning layout – in other words it separates out **/boot** and **/boot/efi**.

Upon login, we discovered that CachyOS still only has a handful of preinstalled apps but they seem to have been carefully selected.

The OS's own *Cachy Browser* is a fork of *LibreWolf*. It also incorporates patches from Gentoo's own version of *Firefox,* as well as "enhanced security and privacy". We noted that this includes bundling the browser with *uBlock Origin* to remove ads and trackers, as well as *CanvasBlocker* to prevent certain types of browser fingerprinting. Both plugins also contain preset rules to protect users when surfing.

CachyOS also maintains its own repositories, which include its own and Arch packages. The OS also has its own *Package Installer,* which serves as a GUI front-end to *Pacman.* On first launch in the live ISO, the package
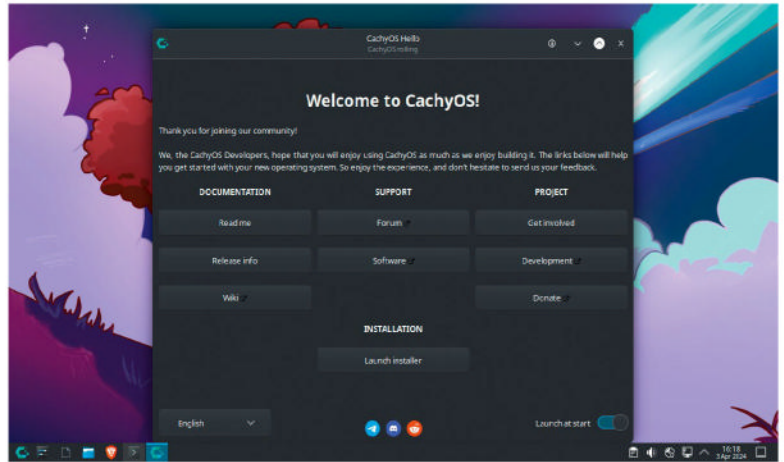
CachyOS's own Hello app opens upon login offering helpful links to the community forum, wiki and project GitHub page.

installer failed to load but told us which *Pacman* command to run in the terminal to fix the issue.

We were then able to open the GUI installer, which arranges available packages into neat categories like Browsers, Games and Mail. We used this to install the *Brave* browser, though were a little nonplussed at having to type `y` to confirm in a graphical installer.

Special mention should also go to CachyOS's *Hello* app, which loads upon first boot. From here you can read the release notes, which is where we learned that version 20240401 is available via two CDNs (content delivery networks) via Cloudflare and Digital Ocean.

From here you can also select Software to launch the CachyOS pkgbuilds page on GitHub to see more information about the distro's own apps. Links are also available for the official wiki, community forum and an excellent Readme with further info for beginners.

During our tests, we didn't notice significantly faster performance than stock Arch Linux. Still, there's no denying that setup is much easier, given the intuitive installer and huge level of online support. **LXF**

VERDICT

DEVELOPER: CachyOS
WEB: https://cachyos.org
LICENCE: Mainly GPL

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

CachyOS truly is Arch made easy. Download to choose from virtually every desktop environment known to man.

» Rating 9/10

Ubuntu 24.04 LTS

Nate Drake goes comfortably numb at the vast number of improvements introduced in Ubuntu 24.04 Noble Numbat.

T his year is already special in that it marks 20 years since Ubuntu first graced Linux users' desktops. April also marks the release of version 24.04 (code name Noble Numbat), the latest Ubuntu LTS (Long Term Support) release.

By default, Ubuntu 24.04 will be supported for five years, until June 2029. This covers all packages in the main Ubuntu repository.

However, Ubuntu Pro subscribers can also benefit from an additional five years of support. Personal subscribers use Ubuntu Pro for free on up to five machines. If subscribers also choose to purchase the Legacy Support Ubuntu Pro add-on, this provides another two years of coverage, effectively meaning that Ubuntu 24.04 can benefit from a total of 12 years of security and support coverage until 2036.

This support applies to the entire distro, not just core packages, which means Ubuntu now offers comparable levels of long term support to RHEL.

Noble Numbat is also available for a variety of CPU architectures. The ARMhf version has received an update to over 1,000 packages, which now use 64-bit over 32-bit values, solving the Year 2038 problem. *(For more information on this bug, read Nate's guide, p.78.)*

We were eager to fire up the latest version, so downloaded the 5.3GB ISO. At the time of writing, this is a daily build, so we encourage your to do your own research into specific features if you read this after the official release date of Ubuntu 24.04 on 25th April.

On first boot, we noted that Noble Numbat now comes with an enhanced installer. Not only is it clearly laid out but it offers the option to configure Accessibility settings for seeing, hearing, typing and using the mouse.

You are also now given the option to choose between Interactive Installation – the step-by-step installation we

Ubuntu 24.04 uses the Gnome 46 desktop. The default Noble Numbat wallpaper can be changed to one of 12 alternative backgrounds.
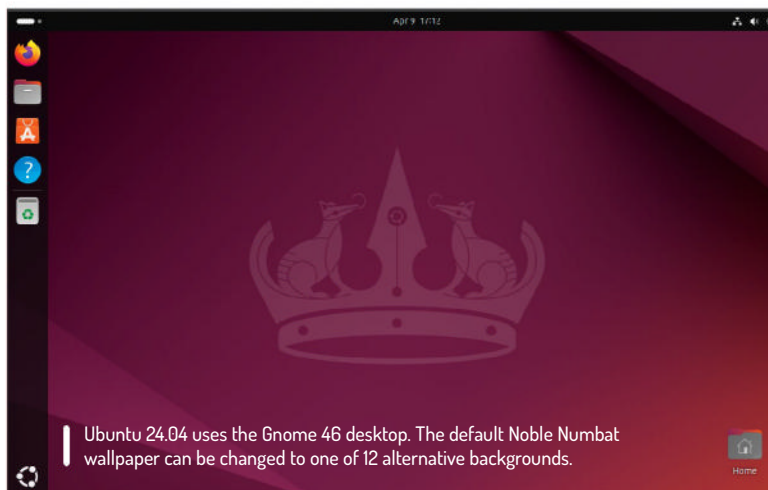
know and love – or Automated Installation. The latter is designed for advanced users with access to an **autoinstall.yaml** file designed to repeat system setup for multiple machines.

Ubuntu 24.04 has reintroduced experimental support for guided ZFS installations, both encrypted and unencrypted deployments. Noble Numbat also carries over its predecessor's tentative support for TPM-backed FDE (full disk encryption) on those devices that support TPM 2.0.

One other important change in 24.04 is that the installer now defaults to the Minimal selection of preinstalled packages. We chose this for our tests, although the installer still allows you to choose an Extended selection of apps including *LibreOffice* and *Thunderbird.* Setup also still offers the option to install third-party software and support for proprietary media formats.

This year Canonical's habitual wallpaper competition has been shaken up a little by including multiple categories such as Digital Art and Nature. This no doubt explains why upon restarting we saw that in addition to the regal mascot background, Ubuntu 24.04 offers 12 alternative wallpapers.

Given that we had opted for a minimal installation, we also took the time to open Noble Numbat's revamped App Center. Sections and apps are grouped closer together relative to Ubuntu 23.10. You can also click into the Manage section to check for and apply any available updates.

Despite the promises in the Discourse section of the Ubuntu website, it seems that plans to display Top Charts of apps have failed to materialise, although, as in Ubuntu 23.10, the App Center arranges programs in an easy-to-find way through dedicated groups, such as Featured Snaps, for example.

When it comes to the apps themselves, users who opted for the Extended installation will note that the

The updated App Center has a more compact layout. Users can click Manage to search for and apply updates.
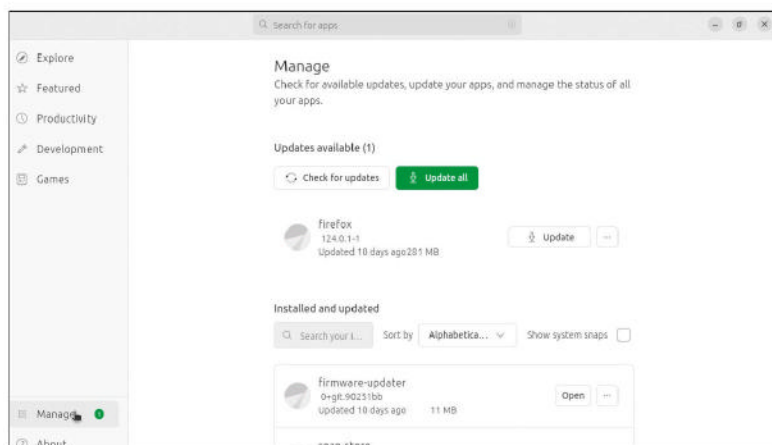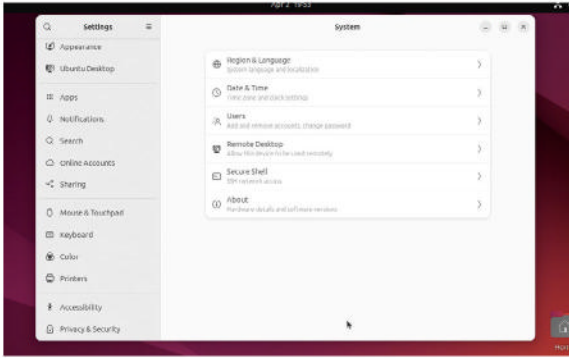
Gnome Settings now includes a dedicated System section to change key settings such as the region/language, date and users.

UBUNTU IN MANY FLAVOURS

Although this review has focused on the default version of Ubuntu 24.04, which uses the Gnome 46 desktop, developers have also been busy on alternatives.

Kubuntu 24.04 is also due for release in April but won't ship with version 6 of Plasma, favouring instead the tried and tested Plasma 5.27 LTS. The OS will also ship with the *Calamares* installer.

The speedy and lightweight Lubuntu OS also uses the *Calamares* installer, with an overhauled Customize section. This offers three different types of installation (Minimal, Normal and Full). The Full option includes third-party packages such as *Krita* and *Thunderbird.* In our tests we found the Lubuntu installer defaulted to the German version but happily switched back to US English.

Despite online claims that Lubuntu 24.04 would come with optional Wayland sessions on the LXQt desktop, when we logged in via a live CD, the system was still using X11.

Xubuntu 24.04 has yet to receive an official release date, though it's likely this will be very shortly after Ubuntu 24.04 on 25th April.

Images are also available for download for the Raspberry Pi 2, 3 and 4. From searching the Pi forums, users have claimed that an overheating issue present on OSes using version 6.7 of the Linux kernel has been fixed in version 6.8, which is used by Ubuntu 24.04.

Visit **https://ubuntu.com/desktop/flavours** to download different flavours of Ubuntu.

popular webcam app *Cheese,* which has been bundled with Ubuntu since around 2010, has now been retired in favour of Gnome's *Snapshot.*

The default version of Ubuntu 24.04 ships with Gnome 46, meaning there's also a number of other exciting tweaks. Some of these are more overt, such as better touchscreen support and a generally more responsive desktop experience.

Other changes are more subtle. For instance, the *Nautilus* file manager now has a search icon at the top-left of each window. Crucially, searches can now be run across your entire system.

The *Settings* app has also been polished, with a dedicated System section in the left-hand pane. This enables users to edit key settings such as date and time, accounts and the system language from one place.

The Privacy section has now been renamed Privacy & Security, although the actual options – such as configuring location settings and purging temporary files – remain the same.

Beneath the hood, Ubuntu 24.04 uses version 6.8 of the Linux kernel. This means Noble Numbat now caches more efficiently due to updated networking buffers and also prevents direct writes to block devices. *AppArmor* now also uses SHA-256 instead of the far less secure SHA1 for policy-hash verification. The kernel also now incorporates the new Intel Xe DRM driver, making Ubuntu 24.04 capable of supporting a greater variety of graphics cards across multiple architectures.

If you've previously played any game on Ubuntu that relies heavily on mmap, such as *Counter-Strike 2* or *Hogwarts Legacy,* you may have run afoul of a bug caused by the fact that **vm.max_map_count** is set to a low value by default (65,530), which can cause games such as this to crash.

The team at Fedora fixed this issue back in November 2023 with the release of version 39 of the OS by changing the default value to 1,048,576. Ubuntu gamers will be relieved to learn that Noble Numbat has now done the same, meaning they're much less likely to encounter this bug in future. The operating system also now ships with version 3.12 of Python.

Special mention should also go to Ubuntu's new dedicated *Firmware Updater,* developed using Flutter. This can be launched directly without having to go via the App Center in order to check for firmware updates to any of your connected devices. Despite ongoing discussions by Ubuntu developers, the *Firmware Updater* isn't grouped under Gnome Utilities, but it can easily be launched via Activities.

The firmware updater enables you to check for updates for any connected devices. For some reason, it's not listed under Utilities.

Despite Canonical's ongoing discussion about sunsetting X11, support remains for X sessions, though Noble Numbat uses Wayland by default.

During our research for this article, we discovered a bug (**https://bit.ly/LXF315ubuntubug**) that currently affects users trying to upgrade from Xubuntu 22.04 to version 24.04. This centres on updating the *Firefox* Snap.

Other than this, though, there seems to be no good reason not to switch to Ubuntu 24.04 today. Noble Numbat offers more than a decade of support, huge security improvements, a slicker App Center and an improved desktop experience. **LXF**

VERDICT

DEVELOPER: Canonical
WEB: https://ubuntu.com
LICENCE: Mainly GPL

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

Version 24.04 proves why Ubuntu has been the distro of choice for so many Linux users for almost 20 years.

Rating 10/10

Roundup

Kali Linux ≫ Parrot OS ≫ BackBox Linux ≫ BlackArch Linux ≫ Pentoo Linux

Michael Reed keeps his ports closed, has passwordless SSH and two locks on his bicycle.

Hacker distros

Donning his best white hat, the always ethical hacker **Michael Reed** examines five Linux hacking distros while hacking into your smart fridge.

HOW WE TESTED…

We began by installing the latest stable release of the AMD64 edition of each distribution into a *VirtualBox* virtual machine. VMs are good for this sort of work as you can make sure that each distro is working within an identical environment. Apart from general use, we tried hacking tutorials using the distros themselves. Where available, we used tutorials and examples aimed at each particular distribution.

All of the distros came with all the pen-testing tools you'd expect to find. There were some differences in exactly what they offered, but you'd have to be at an advanced level for this to make a difference. Instead, we focused on the organisation and documentation for these tools.

This time around, we didn't create a section to test the resource usage of the distributions because, as long as none of them were resource hogs (they weren't), being lightweight isn't especially relevant to this sort of work.

T hey come by different names, such as a hacker, security or penetration-testing distribution, but it all amounts to the same thing. These are distributions that are set up to carry out security exploits in order to test the security of a given system. To this end, they come properly configured for network exploration and testing, and are packed with exploitation tools.

We've found five distributions that can handle every aspect of this type of work, but we've chosen systems that are applicable to beginners who want to get a taste of compsec (computer security) using them.

Kali Linux is the most well known penetration-testing (pen-testing) distribution of all. It comes with hundreds of pen-testing tools and sits on a Debian base, making its workings familiar to the majority of Linux users. Parrot OS (sometimes called Parrot Security) also uses a Debian base but it's got a fairly low-key graphical user interface compared to the others. BlackArch is based on ArchLinux and Pentoo is based on Gentoo. These are both distributions that require a bit of extra Linux know-how to use, and their administration tools aren't quite as well known as those of some more popular Linux distributions out there.

CREDIT: Magictorch

Installation and setup options

How easy is it to get the distribution installed and running?

D ebian-based Parrot Security boots straight into the live desktop, where the installer is invoked from an icon. It's nice to see a distro that embraces Btrfs as the main partition format. Other than that, it's a fairly standard Linux GUI installation experience. Like all of these options, there are instructions on the website to install onto a USB flash drive.

Kali also uses Debian, and the installation is fairly standard but has a couple of extra questions, such as the domain name. There are two options for software choice but it's unlikely that you'd decide not to install the security tools. You can, however, choose between Xfce (default), KDE and Gnome for the desktop. It's also possible to select Expert Install for a text-mode installation that gives more control over installed software and other fine details.

BackBox is based on Ubuntu and the installer is the standard one. The Hardware Information Tool in the GRUB menu at first gives useful information on your installed hardware.

BlackArch uses ArchLinux, and booting from the BlackArch ISO initially takes you to the live desktop. The username and password to log in to this are found on the website, although there was a mistake here as it's listed as `root:blackarch` but accepting the default user (Liveuser) does work. Launching the
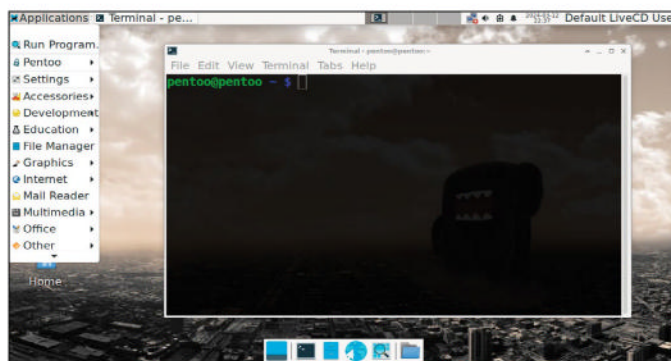
Pentoo uses a text mode installer that offers extra customisation right from the beginning. That said, you can accept the defaults for most of the questions.

installer from the desktop leads to a typical installation GUI but it requires the setup of separate user and administrator accounts. It's also possible to install BlackArch on top of ArchLinux.

Pentoo uses Gentoo Linux. Launching the live CD drops into a text-mode interface to specify language options and password. This leads to the live desktop, where one launches the installer, also in text mode. At various points system configuration options are made available, and some of these are highly technical.

VERDICT

KALI LINUX	8/10	BLACKARCH LINUX	7/10
PARROT OS	7/10	PENTOO LINUX	7/10
BACKBOX LINUX	7/10		

Pentoo was the only option that deviated notably from the standard installation experience. Kali also offers an expert mode installation.

Platform support and editions

Multiple editions for different scenarios and a variety of platforms.

K ali has a huge variety of install options, and these include images for ARM-based computers and pre-made virtual machine images for most of the popular virtualisers. It's even possible to install a version of Kali on to an Android-powered device. There are also containerised versions for low-overhead access to the Kali tools. This is in addition to the expected x86 and AMD64 versions for a normal PC. The website has instructions for making your own custom edition.

BlackArch has three installation ISOs: Full, Slim and Netinstall. We went for Slim as the Full edition (22GB!) is only useful for installation without an internet connection. Netinstall installs a basic system, leaving it to you to add the optional tools manually. There are instructions to add BlackArch packages to standard ArchLinux but it doesn't give quite the same distro as BlackArch.

This aspect of the website is confusing, but as far as we can see, Pentoo offers x86 and AMD64 versions in either beta or stable versions.

There is a donate option on the BackBox website but the online instructions inform you that you can set this to zero if you prefer. There is only a single edition, for X86-64 PCs.

Kali Linux has pretty much every angle covered when it comes to platform support. This includes hardware platforms, along with virtualisers and container formats.

Parrot OS also has a good variety of installation candidates to cater for platforms such as *Docker* containers, popular virtualisers and the Raspberry Pi. The Architect Edition comes with very little installed and includes a menu-based system for adding system components and different types of hacking tools.

VERDICT

KALI LINUX	10/10	BLACKARCH LINUX	6/10
PARROT OS	8/10	PENTOO LINUX	6/10
BACKBOX LINUX	5/10		

Kali is the leader in terms of the sheer number of supported platforms and specialised editions, and Parrot Security offers a lot, too.

The desktop experience

How does it look and feel in use?

L et's get something out of the way from the beginning – some of these desktops adopt a 'hacker' aesthetic. There's nothing wrong with this in itself, as long as it doesn't get in the way of normal functionality. There's even something to be said for the desktops that don't bother with a hacker look because it means that they are easier to use unobtrusively. We're not going to mark up or down either way. Just bear in mind that some of these desktops make it look as though you're hacking something.

These distributions are designed to be occasional or secondary use systems rather than daily drivers, so we prefer to see a standard desktop and window layout to something that has to be learnt from scratch or that might be confusing when first encountering it. They aren't daily drivers, but we expect these desktops to offer all of the usual facilities, such as a full file manager and a searchable application launcher that pops up when we press the Super key.

Kali Linux   9/10

Gnome and KDE are options, but Xfce is the default Kali desktop. This has a dark theme and some dark, shiny graphics on the backdrop. So, it evokes a hackerish mood without being a total giveaway that you're doing some hacking. The taskbar is at the top, meaning the searchable application launcher is a pull-down one.

We like the items placed on the taskbar as they include a switcher between four virtual desktops and, most importantly, some monitor graphs for CPU usage. We like that, but it's a shame that network usage monitoring isn't also set up. It can be added, though, as with any Xfce desktop. The drop-down list of useful folders is another sensible addition.

The terminal console has a customised layout with good use of colour.

Overall, the Kali desktop is a sleek-looking dark setup with lots of useful features rather than being dedicated to hacker visual bling.
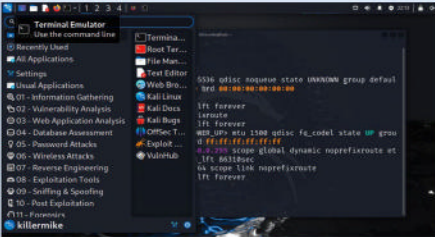
Parrot OS   8/10

When booting, you're treated to an animation with a glitchy effect, reinforcing the hacker aesthetic, but the desktop uses an attractive dark, flat, clear theme rather than an obviously hackerish one.

Parrot Security uses the MATE desktop environment, a continuation of Gnome 2. There are two app launchers, a pull-down menu-type set of launchers (Application, Places and System) at the top and a more typical, searchable launcher at the bottom that pops up when you press the Super key. It has CPU, memory and network monitors in the top taskbar.

The window control icons – red, green and yellow blobs – have a slightly non-standard placement. Having an icon on the bottom bar to toggle automatic sleep mode is a nice touch.

The terminal shell is customised and uses a hackerish green font and places the username and current directory above the entry line, a useful layout.
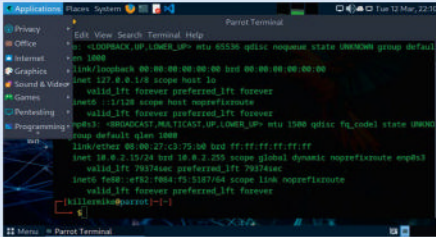
The reading stuff!
Instructions to get the distribution up and running and to use it.

T he official Kali documentation is split into small sections and covers most of the topics you're likely to be interested in, ranging from installing Kali to more advanced topics such as installing *Docker* in the distro. Kali offers an official training course that features a certificate on completion. There is a separate page with summaries of the included security tools with links to their respective pages.

We didn't think that anything would beat the Kali documentation, but the Parrot Security website takes a similar approach in terms of splitting the docs up into small articles, but there seem to be slightly more articles and they tend to be a little longer.

On the Pentoo website, clicking on the Docs link leads to a page with a small amount of information about boot options and a nine-year-old YouTube video about the project. There is also a brief FAQ. The documentation is underwhelming and it's obvious that the developers expect the user to rely on Gentoo documentation and forums.

The BlackArch Linux site has a manual in PDF format. It covers installation, but much of the rest of the manual is effectively empty as it just says "Coming soon…". The site does have a long list of all included tools with descriptions and links to the developer websites, though.

The BackBox site was something of a mess in this regard. The documentation link threw up a broken wiki link – a Telegram post seemed to indicate the wiki is defunct. Unless finding your way to the documentation is some weird meta-test that we failed, getting it fixed would be our advice.

VERDICT

KALI LINUX	8/10	BLACKARCH LINUX	6/10
PARROT OS	9/10	PENTOO LINUX	3/10
BACKBOX LINUX	0/10		

The Kali and Parrot OS projects come with professional-looking comprehensive documentation. The other projects were less impressive.

BackBox 6/10

Xfce powers the BackBox desktop. Along the top, we have a taskbar with a launcher, decorated with the BackBox logo in the corner. This launcher is the standard one, but it's the searchable version and it gets the job done, even though it's not configured to launch when you press the Super key. Overall, this is a fairly standard Xfce desktop, including the weird, razor-thin sensitive areas at the edges of windows that take ages to grab.

Like most of the rest of this desktop, the terminal is the stock one that we're used to seeing. Most users will probably have to increase the font sizes in the terminal, if not the desktop overall.

Put everything together, and it's a functional desktop that doesn't deviate too far from Xfce defaults. This means that it lacks the l33t hacker pizzazz of the other desktops, but it has to be said that some of the others also have some useful practical additions, too.
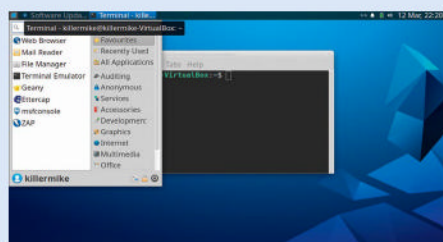
BlackArch 8/10

When we first clapped eyes on the desktop, we thought we might be looking at something using Plasma, but it's actually a dark and transparent Xfce desktop, and very sleek and hackerish it is, too. However, it struck us as funny because the backdrop is covered in code excerpts in a tiny font that are purely decorative. But, if it makes you feel more like a 'hacker', then good for you.

On repeated use, we found that BlackArch is set up so the applications that were running in the previous session are launched again when rebooting, which we'd rather not see enabled by default.

In terms of window decorations, the minimise, maximise and close icons are Mac-like coloured blobs, but at least they are in the standard location, meaning only a short period to acclimatise. As with many Xfce desktops, the sensitive areas of a window edge are too thin. Overall, it's an attractive hacker-style desktop.

Pentoo 8/10

As it uses the Tango icon set, we first guessed that this was a Gnome-derived desktop, but Pentoo is yet another distro that runs on a customised Xfce desktop. The taskbar runs along the top, but the main launcher can't be searched. There is a separate searchable launcher, though, accessible on the icon bar, but it doesn't launch when you press the Super key.

Thunar is a common sight on this type of distribution, and it's included here – we're not surprised, as it's lightweight, fast and includes a network browser. The CLI has colour enabled, but it's a fairly stock terminal configuration.

It's a nice contrast to see a smart, white business-like desktop on this type of distro for once. Unlike some Xfce desktops, there is a reasonable area of sensitivity around the edges of windows, making them easy to grab when you have to. Overall, it's a pleasant, functional desktop setup.
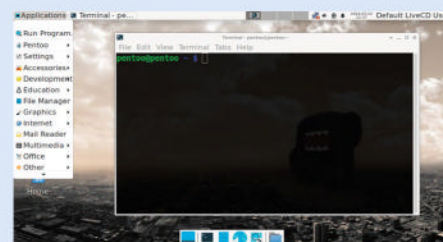
Forums and other support

Vitally important when learning the system or if you run into trouble.

Kali has a built-in advantage due to its popularity, meaning that tutorials are available around the internet and many penetration tutorials assume that you're using Kali. This helps when you're first learning to use the distro. Being a well-known distro means that external resources such as the Reddit subreddit (r/Kalilinux) are extremely active.

There is an official Kali forum. It has moderate activity and suffers from having too many sections. However, it looks as though every question that's posed gets a reply. A search for Kali Linux on YouTube reveals dozens of videos and some complete training courses.

The Parrot Security official forum has similar levels of activity to that of Kali, but the layout is better for a moderate traffic forum. As with Kali, posts always seem to receive an answer. YouTube showed a few videos that mention Parrot Security, and the Reddit subreddit has a low but steady level of activity.

On the BlackArch website, links to community resources don't seem to work properly, and we couldn't find an official forum. The Reddit subreddit has a few posts a month, but they do tend to have comments underneath.

Various parts of the BackBox website look as though they need to be updated. The BackBox forum is largely abandoned, but the Announcements subforum is still occasionally posted to. BackBox also has a Telegram group with 525 members.

Pentoo doesn't have any community links, such as a forum, on the website. It is clear that you should post questions about Pentoo to the Gentoo forum, although Pentoo doesn't have its own section there.

VERDICT

KALI LINUX	9/10	BLACKARCH LINUX	4/10
PARROT OS	7/10	PENTOO LINUX	4/10
BACKBOX LINUX	4/10		

Kali is the undisputed leader of this group when it comes to official and unofficial support resources.

Specialist tool organisation

With hundreds of tools, it helps if you can find the one you need.

Kali works differently from the others as it has the security tools in the root of the app launcher. If you know what you're looking for, the whole thing is searchable from the launcher. The 'normal' apps are located in the Usual Applications folder, and sure enough, the security tools are absent from here. The security tools are organised into 13 numbered categories, many of which have subfolders.

As with all of these distros, Kali has a mixture of command-line and GUI tools, and if you click on a command-line tool, it opens a new terminal window. However, it's not always obvious at a glance whether you are dealing with a GUI application as many of the command-line tools also have an icon. None of the distros handle this perfectly.

On BlackArch, the searchable application launcher is well categorised with an expandable set of subcategories under the BlackArch category. There are a few extra tools (such as *Wireshark)* scattered around the more general categories, something we'd prefer not to see.

BackBox puts all of the security tools into the first three categories: Auditing, Anonymous and Services. These contain the expandable subfolders.

Parrot OS comes with 600+ tools. Good, searchable organisation is a must. Like all of the distros, we found it difficult to distinguish between command-line and GUI tools.
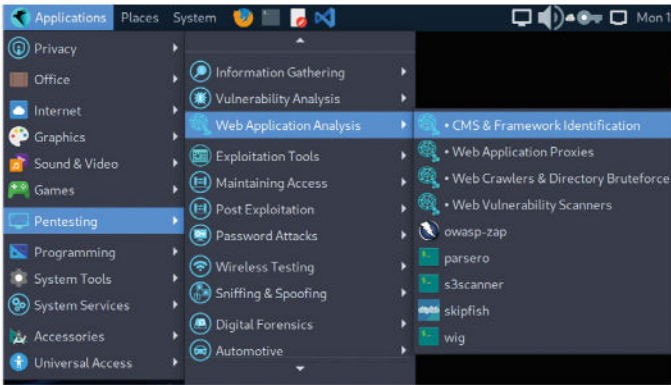
Most of Parrot Security's specialist tools are in the Pentesting category. However, there are no subcategories, and dozens upon dozens of tools are presented as a huge list. There is some categorisation for other tools, though.

Pentoo's specialist tools are well organised into categories, and the separate launcher on the icon bar presents a searchable list of all of the applications on the system.

VERDICT

KALI LINUX	**8/10**	**BLACKARCH LINUX**	**7/10**
PARROT OS	**5/10**	**PENTOO LINUX**	**7/10**
BACKBOX LINUX	**7/10**		

Kali has the greatest level of consistency of tool organisation, while Parrot OS has some problems with organisation.

Suitability for beginners

If you are beginning your compsec career, can you get started?

These distros are focused on a technical area of computer use, penetration testing, but that doesn't mean they're not suitable for beginners in this discipline.

It's not much of a surprise that a Gentoo-based distro like Pentoo is more complicated to use. Even installation requires some extra know-how. When Pentoo is booted, it doesn't automatically start the graphical environment, and the network is disabled. The icon to enable networking wasn't added to the desktop once we'd carried out an HD installation (a bug?), so we had to look online, on another computer, to find the command to get networking going. Gentoo is great, but it's less well known than the Debian or Red Hat-derived distros, so the average Linux user would have to learn how to use the package management and other administrative functions that are specific to Gentoo.

ArchLinux (and therefore BlackArch) also assumes a certain amount of technical knowledge from its users, but as it's not a source-based distro, it's not quite as complicated to use as Gentoo. Having said all that, if you have only used Ubuntu, for example, you would have to learn quite a bit to administer an ArchLinux-based system. For penetration testing, you have to do
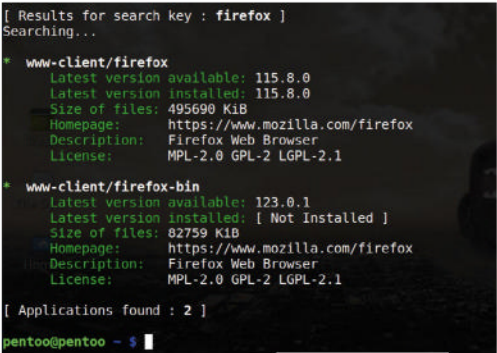
Gentoo-based distros work a bit differently from those derived from Debian or Red Hat in the areas of package management and system configuration, and tend to be a bit more complicated.

more work than usual in this area. Although, some would see this as an opportunity to learn something new.

Kali is Debian-derived, so you'll understand how to operate it from the command line if you've used a distro such as Debian, Ubuntu or Linux Mint. We could make the same observation about BackBox and Parrot OS as they, too, are Debian-like. Kali's popularity means that many pen-testing tutorials assume you're running Kali. The excellent documentation has to be taken into account, and we could say the same for Parrot OS, too.

VERDICT

KALI LINUX	**9/10**	**BLACKARCH LINUX**	**5/10**
PARROT OS	**8/10**	**PENTOO LINUX**	**5/10**
BACKBOX LINUX	**8/10**		

Kali has the tutorials and typical desktop Linux ease of use. Parrot OS and BackBox are also similar to typical desktop Linux.

The verdict

Hacker distros

K ali has it all if you want a pre-made network-hacking distribution. It has an impressive number of editions to cover most installation scenarios, plus a slick-looking lightweight desktop and a huge set of built-in, well-categorised hacking tools. It also benefits from being the most popular hacking distro, so there are plenty of tutorials and many penetration-testing tool tutorials use Kali as a base. We suspect that the average Linux user could have Kali installed and be following a penetration tutorial within a morning if they already have some experience with networking tools.

Parrot OS gives Kali a run for its money in most aspects as it features a stylish desktop, a comprehensive set of built-in tools and a decent variety of installation editions, too. For beginners, it's worth considering that it runs on a Debian base, like many popular Linux distros.

We've looked at these distros from the perspective of a beginner to intermediate user. This diminished the advantages of the ArchLinux and Gentoo-based distros as they are more readily customised by an advanced user.

BackBox is a wolf in sheep's clothing as it has the most restrained-looking desktop, but as normal-looking as it is, it has a full complement of security tools.

BlackArch used about half the RAM on boot of a typical distro, while offering a modern-looking desktop. It's based on ArchLinux, so you'd have to familiarise yourself with the administration tools of ArchLinux if you're not used to working with that distro and its derivatives. Considering the type of work you tend to do with a hacker distro, take into account that you'll probably have to do some admin work in terms of adding packages and configuring network services. However, ArchLinux is well supported when it comes to forums and other help.

Pentoo is based on Gentoo Linux, which automatically adds a bit more complexity in use in exchange for quite a lot of extra customisation capability. Like BlackArch, on boot, the memory usage was lower than the typical distribution. However, as it's based on Gentoo, much of the expansion of the system through package management is slower than a typical distribution, as it involves compiling source code. Like ArchLinux, Pentoo's base distro Gentoo is a well-supported distribution.
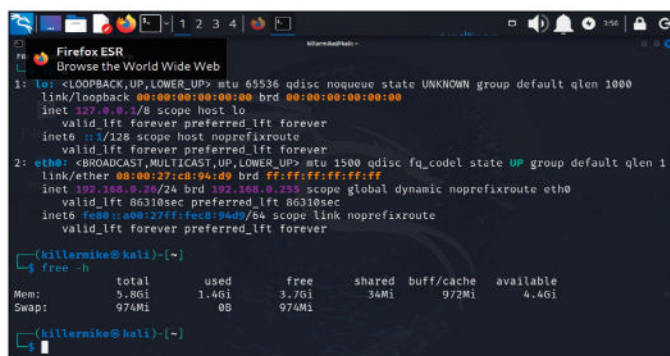
1st	**Kali**	10/10

Web: www.kali.org **Licence:** **Various open source**
Version: **2024.1 (February 2024)**
Well supported, officially and around the net. Easy to get started with.

2nd	**Parrot OS**	9/10

Web: https://parrotsec.org **Licence:** **Various open source**
Version: **6.0 Lorikeet (January 2024)**
Good documentation and slick looking. Similar to other Debian derivatives.

3rd	**BackBox**	8/10

Web: www.backbox.org **Licence:** **Various open source**
Version: **8.1 (November 2023)**
Normal-looking desktop. Typical Linux desktop ease of use.

4th	**BlackArch**	7/10

Web: https://blackarch.org **Licence:** **Various open source**
Version: **2023.05.01 (May 2023)**
Not as suitable for beginners. Cool-looking desktop. Highly customisable.

5th	**Pentoo**	7/10

Web: www.pentoo.ch **Licence:** **Various open source**
Version: **2024.0 (February 2024)**
One for the experts. Highly customisable from installation onwards.

›› ALSO CONSIDER

ArchStrike (**https://archstrike.org**) is, like BlackArch, a pen-testing distro that uses ArchLinux as its base. Unfortunately, it doesn't seem to have been updated since 2021.

Caine Security (**www.caine-live.net**) comes with tools for careful analysis of storage media and files. For one thing, all devices are mounted as read-only until the user deems otherwise. Generally speaking, we'd rate it as a complicated distro to operate, and its documentation is inconsistent. It's based on Ubuntu and it hasn't had an official release since early last year, which was another reason it wasn't included.

Network Security Toolkit (**https://wiki.networksecurity toolkit.org**) is based on Fedora, so uses package management and sysadmin tools common to most Red Hat-derived distros. The website has a well-stocked wiki and other documentation resources. Fedora Security Lab (**https://labs.fedoraproject. org**) is another Fedora-derived security distro. **LXF**

Upgrade to
Ubuntu 24.04 LTS

The latest Ubuntu is here. **Jonni Bidwell** has come out of hibernation just to celebrate and (in between beverages) tell you all about it.

CREDIT: Magictorch

Ubuntu LTS (Long Term Support) releases are one of the few things we get excited about nowadays. So colour us excited. We're champing at the bit to tell you all about Ubuntu 24.04's great new features. And to explain the creature behind its code name (Noble Numbat). As ever, Ubuntu continues to excel at usability, compatibility and utility. Gone are the days when gripes about *Office* documents and Windows games reigned. Whether you're a beginner or aficionado, we've got all you need to get started.

We'll show you how to get it installed quickly and safely. We'll show you how to make sure it's playing nice with your hardware. We'll show you all the fresh features, such as the brand-new Gnome 46 desktop and accessibility options. We'll show you how to install powerful open source software. We'll even show you how to install proprietary games. Yes, we have a lot to show you. And if there are things you don't like, we might also be able to show you how to remedy them.

We've got the low-down on Snaps, the universal packaging format from Canonical (the company that makes Ubuntu). They've been getting a bit of flack lately, so we're going to counter that and show that they're actually pretty good. We'll show you how you can use them to get powerful software installed effortlessly. And how they use confinement to keep you safe. There's a whole lot to get through, so grab yourself a brew (or something stronger) and let's see what the Numbat is all about.

Make way for nobility

Prepare yourself, mentally and digitally, for Ubuntu's latest and greatest outing, Noble Numbat.

I t's that most special time of the year when Canonical releases a new Ubuntu LTS. This time, the creature it has loosed unto the world is Noble Numbat. And what a fine specimen it is. OK, full disclosure before we get stuck in: we're basing our evaluation on the latest daily images. So, we're ignoring some rough edges as we're sure they'll be smoothed by the time you read this. This does mean we may miss any final flourishes the team is holding back until the final release, but we're sure you'll forgive us.

Before you rush to **https://ubuntu.com/desktop** to download the installation material, we'd like to offer a few pointers. If this is your first foray into the world of Ubuntu, or Linux in general, good news: we're making this feature as beginner-friendly as possible. The flipside of this is that installing Ubuntu is sometimes a bit trickier than actually using it. Ideally, you'll want to install Linux on its own machine. This way, you sidestep any issues that arise from dual booting with Windows. Make sure there's nothing important on that machine; if there is, back it up right away. The easiest way to install Ubuntu is to let it erase an entire drive, and once that's happened, you're not getting anything back.

Not everyone has a spare machine (although pretty much any 64-bit system with 4GB of RAM can happily run Ubuntu, so you might want to cobble something together). The next best thing is to get a fresh new SSD (256GB is more than enough), chuck it in your system and dual boot without touching the Windows drive. Windows 10 was fussy about having its partitions and
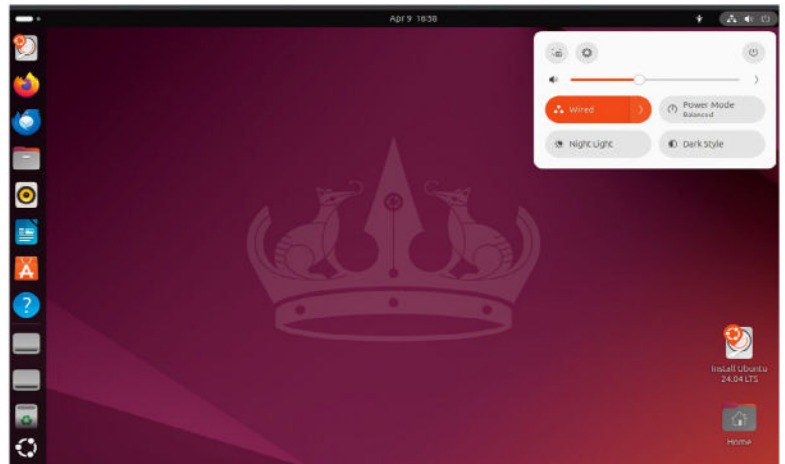
filesystems resized, and Win 11 is even more so. So, we caution against trying to get the two OSes to play nice on the same drive, even though it's theoretically possible to do that from the Ubuntu installer.

Windows 11 requires Secure Boot to be turned on, but (unlike the **LXFDVDs** of old) Ubuntu has no problem with this. There's no accounting for other oddities in manufacturers' UEFI (the firmware that runs as soon as you turn on your machine) implementations, so all kinds of things could happen (or not happen) on that critical post-installation reboot. While these are usually not indicative of massive data loss, it's vital that you back up critical files from your Windows drive in case.

Ubuntu lets you sample the distro (or wreck it completely) without touching your system, using live media.

» CREATE AND BOOT INSTALLATION MEDIA

You can download the Ubuntu 24.04 ISO image from the link above. It's over 5GB, so won't fit on a regular DVD. Unless you've got a thing for dual-layer optical media, you'll want to write that image to a USB drive. This is more complex than just copying the file on to it, and you need specialist tools to do it. If you're already running Ubuntu, it includes its own tool for this, called *Startup Disk Creator*. Otherwise, we recommend *Etcher,* which you can get from **https://etcher.io**. It runs on Windows, Mac OS and

Linux. Fire up *Etcher*, select the ISO you just downloaded and hit Flash. Now make a cup of tea while it's writing.

Exactly how to boot your freshly minted media depends on your system. Some give you a helpful boot menu when you push a key (often F12 or F10) while booting. Others require you to explicitly change the boot order from the UEFI configuration (which you might get to by pushing Del or Alt+F2). In some cases, it's tricky (if not impossible) to catch the UEFI before Windows boots. In this case,

Etcher makes easy work of creating Linux boot media.
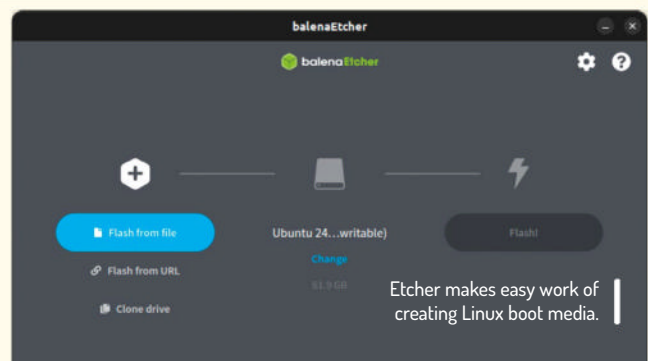
you can force the issue from the Windows login screen (or Start menu). Hold down Shift as you click the power icon in the bottom-right (or Power Options in the Start menu)

and select Restart. Then from the recovery menu, select Advanced Options > UEFI Firmware Settings. Dig around in here and you'll find some boot options.

Install Ubuntu 24.04

Get Ubuntu loaded on to your machine, without messing with any incumbent operating system.

H opefully, by now you've managed to create and boot an Ubuntu installation medium. If not, turn back and follow the instructions in the box forthwith. The first thing you see is a brutalist menu from which you should select the first option (Try Or Install Ubuntu). If things go wrong, try again with one of the other options (use Safe Graphics mode if you see graphical corruption or a blank screen). Once it's loaded, you hear a welcoming chime and are asked to select localisation settings, connect to your Wi-Fi (or LAN), and set up any accessibility options (such as the zoom tool, screen reader and sticky keys). You might then be offered an update to the installer, which you should accept.

Next you're asked whether to Try Ubuntu or Install it. Rather than dive straight into the installation, we first recommend trying out the live environment. It doesn't touch your system drive(s), unless you ask it to – so it's totally risk-free, unless you take risks. The live environment runs from a compressed filesystem and without much in the way of graphical acceleration (depending on your hardware), so it won't quite be as snappy *[is this foreshadowing? – ed]* as the real thing.

If you're coming from a Windows background (in which case, congratulations on taking this first step), the most obvious difference is the lack of a Start menu equivalent. The flagship edition of Ubuntu uses the Gnome desktop, which, as far as GUIs go, is as far from Windows as you could get. There are other Ubuntu flavours that use other desktop environments (some of which have a traditional applications menu) – more on those later. In Ubuntu, frequently accessed programs are accessed from the Dock to the left. If you have a

You might need to adjust display scaling if you have a high-resolution display.
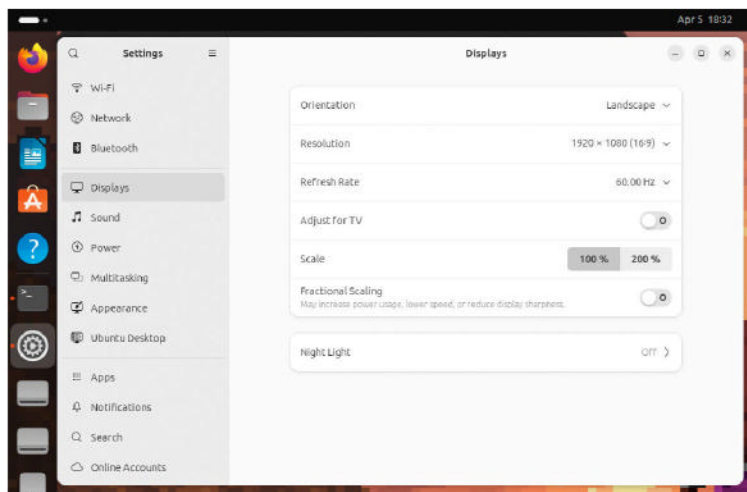
high-resolution display and everything looks too small, right-click the desktop, select Display Settings and then adjust the scaling. Your first Linux experience should not involve eye strain, after all.

Dock worker

The live environment includes (in order of Dock appearance) *Firefox,* a file manager *(Files),* the word processor from *LibreOffice (Writer),* App Center for installing and updating software *(see box)* and the *Help* tool. Below this are any connected drives and the Trash (where deleted files first end up). That's not the whole story, though. Get a list of all installed applications by clicking the Ubuntu logo at the bottom of said Dock. If you dig around in here you'll find the rest of *LibreOffice,* the *Gparted* disk partitioning tool (in case you want to prepare your disks for a more advanced install) and a myriad other utilities that it would be silly to list here. Basically, everything you need to get started.

You're probably already familiar with *Firefox* (it looks and works exactly like it does on Windows and Mac OS) and, for most purposes, *Files* acts much like Windows Explorer or the Mac OS Finder. If you're installing alongside Windows (and your Windows drive doesn't use any kind of encryption), you can see those files, too. The indicator icons in the top-right show network, volume and battery status. You can adjust decibels by hovering the mouse over the indicator icons and using the scroll wheel. If you click on them, you bring up the Quick Settings menu, which can toggle Bluetooth, network and light/dark themes. You can also configure Night Light here, to automatically reduce blue hues after sunset, again reducing eye strain.

From the top-left of the Quick Settings menu, you can bring up the screenshot tool (invaluable for tech

Accessibility is now at the heart of the Ubuntu setup, making it (er) accessible for all.
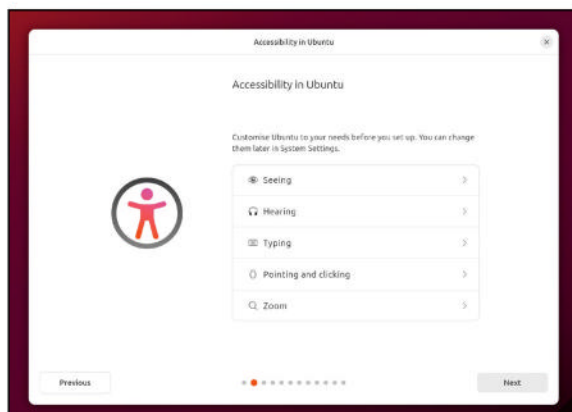
journalists). The small cog immediately to the right of this brings up the Settings panel, from whence you can configure all aspects of the system. We'll leave you to rummage around in there – it's a tweaker's paradise. The Status menu also provides the buttons to shut down or restart your machine. But let's not do that; we've nearly completed our cursory introduction and are but a few sentences away from installing.

If you click in the top-left, you get to the Activities overview. This shows you any running apps, and also lets you switch between workspaces. These are virtual desktops that enable you to better organise yourself when running multiple apps. If you make heavy use of these, you'll want to know that the Windows key (technically the Super key) brings up the Activities view, and that Super and Page Up/Down switch between workspaces. If you type in the first few letters of a program from the Activities view, it (or any other app matching those letters) magically pops up, and you can open it by clicking or hitting Enter.

Get it installed

When you're ready, close any open applications and click the Install Ubuntu icon on the bottom-right. You're asked the same questions as earlier, so zip through those until you're asked whether to do an Interactive or Automated installation. The correct answer is Interactive, so click it. Next you're asked if you want the Default or Extended package selection. The Default installation is somewhat frugal (basically, the apps we mentioned above, minus *LibreOffice).* The Extended selection includes *LibreOffice* and a few other utilities. Size-wise the Extended install occupies 9.4GB…

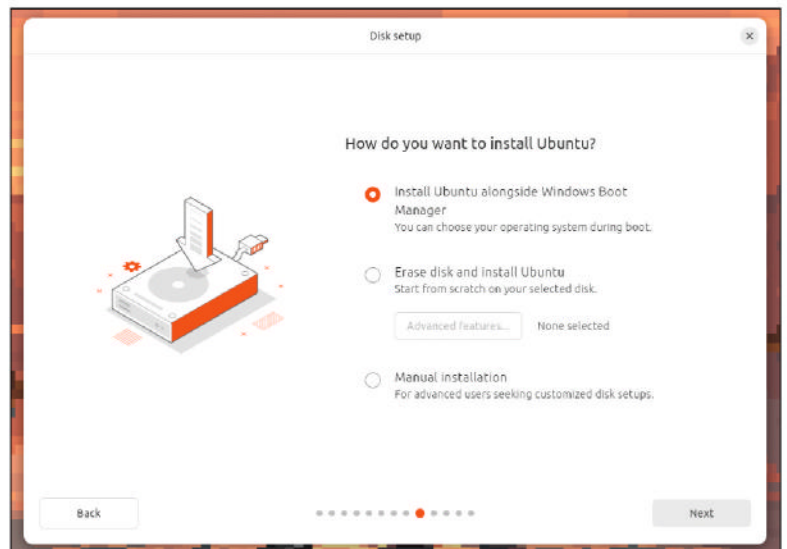Next you're asked about proprietary software. This isn't the place to enter into a philosophical discussion on software licences, so if you're not familiar with the lore (and particularly if you have an Nvidia graphics card), tick the first box. It might also be the difference between your Wi-Fi working or otherwise. The second box is less important, but if you still use the MP3

Ubuntu happily installs alongside Windows, but Windows might not be happy about it being there.

format for your music collection (or Apple QuickTime), tick it. Now you're asked whether you want to erase a disk and install Ubuntu (if it detects a suitable drive), install alongside another OS or go with a manual install. Choose whatever suits, but if you're dual booting, make sure the correct drive is marked for installation. If Ubuntu detects files or an OS on that drive, it tries its best to repartition it sensibly and it will probably be fine. But we'll reiterate our advice from earlier – it's much safer to devote a whole drive to Ubuntu.

Once you've chosen your installation method, you are asked to set up an account with a username and password. You're asked for this password when you log in, unless you uncheck the box below. There's also the option to log in via an Active Directory domain, but we doubt any of our readers use this at home. All that's left to do is wait for the install to complete. Enjoy the slideshow showcasing what awaits, or make a cup of tea. When done, choose Reboot and, when prompted, remove the installation medium and hit Enter.

» INSTALL SOFTWARE

Besides the lack of Start menu, one of the most stark differences between Ubuntu and Windows is the way software (be it applications, drivers or games) is installed. While this is changing on Windows, many users are still used to downloading software from (sometimes questionable) websites in the form of an executable (or compressed executable) bundle. While things on Ubuntu (and other flavours of Linux) can be installed this way, the de rigueur way to install software is direct from Ubuntu's repositories. This way, you can be sure that

what you're getting is what it's supposed to be, and it will work with everything else installed there.

The App Center is your gateway to these repositories, so let's explore how the process works by installing something. One change Canonical has introduced for this outing is that *Minesweeper* and *Solitaire*-type games are no longer installed by default. Let's change this by installing *Gnome Mines.* Open the App Center, search for `gnome mines` from the top bar and hit Install. Within moments, that button changes to read
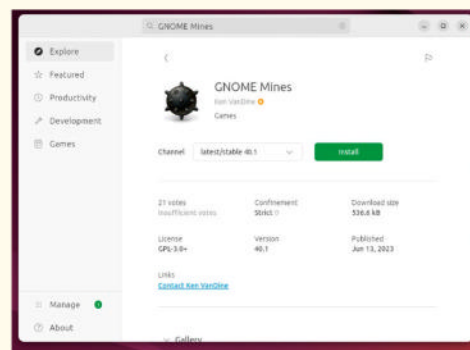
Bring back Mines (and other classic Gnome games) in a couple of clicks.

Open and you can waste some time finding explosive ordnance. Again, this is all happening in the ephemeral live environment, and isn't making any changes to your system (or the USB you're

installing from). So, feel free to install anything else, just bear in mind there isn't much space in the live install (it's a 2GB filesystem that lives in memory). So, you can't install anything heavy like *Steam.*

Explore Ubuntu

Get to grips with your new favourite operating system and diagnose any hardware or software issues.

You should reboot to a boot menu (which is quite hard to read on a HiDPI display). Choose Ubuntu (it's the default option, with Windows Boot Manager below it if you installed alongside Windows) and in a few seconds, you are greeted with a login screen. Enter your password and you're presented with a prompt about sending information to Canonical. Unlike certain other OSes, Ubuntu cares about your privacy. Any telemetry information is not linked in any way to your machine or identity. It's easy to opt out of sending anything, but if you want Canonical get some insight into how Ubuntu is used or how it's breaking, say Yes. You can even view the first report before it's sent to get a handle on what kind of metrics are divulged.

Next you're asked if you want to enable Location Services, which allows apps to guess where you are based on your IP address, GPS signal and nearby Wi-Fi/Bluetooth devices. This was done using the privacy-friendly Mozilla Location Service, but in March it was announced that this service is shutting its doors to non-Mozilla entities, and at the time of writing, it's unclear how Ubuntu plans to mitigate this. The most likely scenario is that this will reduce to a simple IP-based lookup, but we won't speculate. If you object to being located, say No, then say hello to your fresh new Ubuntu desktop. A desktop that, but for an installation icon, is indiscernible from the live environment.

If you are anything like us, you'll want to check out the selection of wallpapers. Ubuntu has always shipped
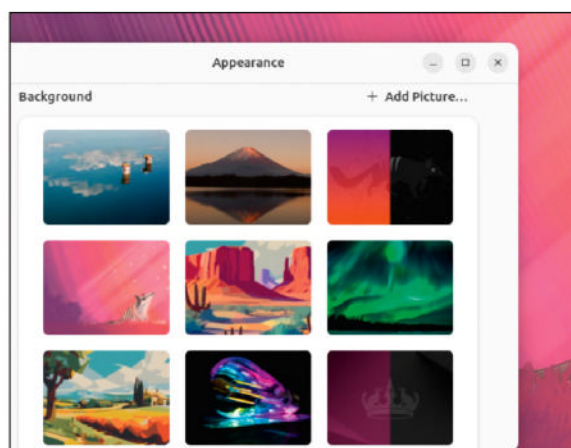

The best part of any new Ubuntu release is the fantastic selection of desktop backgrounds.

with fabulous desktop background offerings, and 24.04 is no different. As usual, some of the graphics are based on the animal (a numbat is a termite-eating marsupial found in Western Australia, if you didn't know) and some are just nice to look at. Right-click the desktop and choose Change Background to see for yourself. You might also have your own images you want to use here, in which case just select Add Picture from the top. It's a controversial option, but you might also want to set Bing's image of the day as your background, in which case you can install the *BingWall* application

» THE TERMINAL

Ubuntu is used by all sorts of people, from beginners to pros, business suits to scruffy gamers, and everyone in between. Whatever your level or purpose, if you want to deepen your understanding of Linux (or possibly just fix some annoying problem), then sooner or later you're going to have to face the command line. This is a text interface to your system (accessed by the *Terminal* application), which in the right hands, can manipulate every aspect of it. If you're a Windows pro, then think *Powershell*. Or if you're old-school, then think DOS on steroids. There's no space for a deep dive (*see* **https://bit.ly/lxf_terminal**) into shell commands here, but if you type `journalctl -b` you'll see the system log for the current boot. Use the cursors to scroll up and down, and you might find some clues to help diagnose any problems. Or you might find some problems you never knew you had. Press Q to return to the command prompt.

Other useful commands are: `top` (see a list of running processes), `df -h` (show free space) and `lspci` or `lsucb` to show a list of connected PCI/

No one can tell you what the command line is, you must experience it yourself.

USB devices. By far the most quoted in this magazine, though, is the incantation:

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

This invokes administrative privileges (the `sudo` part of the commands) to update package lists and then apply any available upgrades.

from the App Center (sorry fellow Brits – this is how it's spelled).

If you're using a laptop, it's worth noting that you can switch between three power profiles from the Status menu: Performance, Balanced and Power Saver, which we're pretty sure you don't need us to explain further. In the latest release, support has been reworked to make newer hardware (where P-state drivers are available) even more efficient when working on battery power. All sorts of hardware (besides the CPU itself) can be power-managed these days, including graphics, storage and network devices.

If things go wrong

Ubuntu is pretty good at getting all your hardware working out of the box. The days when you had to configure each bit of hardware through arcane text files are mercifully (for the most part) far behind. That notwithstanding, if you feel something hasn't been set up correctly, it might be worth investigating the *Additional Drivers* utility (hit the Ubuntu logo in the bottom-right to bring up the applications menu and search for it there). This tells you if any proprietary drivers are active or can/should be activated. As we touched on, you shouldn't ever have to manually download and install any drivers on Linux. And while some manufacturers do offer Linux downloads, these are often more trouble than they're worth.

If you run into a showstopping graphics issue, you might find yourself staring at a textual login prompt instead of the usual pretty graphical one. Push Ctrl+Alt+F3 to see what we mean if you don't have a graphics problem (Ctrl+Alt+F2 brings your desktop back). Your usual login details work at the console, and after entering them, you meet an expectant-looking flashing cursor. Welcome, friend, to the console. We've confined all we want to say about this to the box *(opposite),* but if you run into graphics or other problems, we encourage you to take your error messages and log files to **https://ubuntuforums.org** or **https://askubuntu.com**. Just be sure to read the guidelines before posting.

It's likely (if you're connected to the internet) that a couple of minutes after logging in, you're notified that updated software is available. Like any modern OS, you should update Ubuntu regularly. Unlike Windows,

Ubuntu does not suddenly force updates upon you, nor does it make your machine unusable for hours while they install. Update alerts (and indeed any notifications) can be muted by toggling the Do Not Disturb switch in the notification area. They can also be configured on a per-application basis from the Settings panel.

If you get an upgrade alert, you might notice some updates are marked as Ubuntu Pro updates and greyed out. Ubuntu Pro is a little controversial (among free software users) because it's (potentially) a paid-for service that seems to gatekeep certain updates from non-Pro (amateur?) users. In truth, there's no real need for this fuss. Ubuntu Pro is free (on up to five machines) for home users, and if you don't want it, you're not going to miss any vital new features and you're certainly not going to be deprived of any critical security updates. If you want to activate it, you need to sign up for an Ubuntu One account at **https://login.ubuntu.com**. No credit card details are taken (only a valid email address is required). Once you're enrolled, you can activate the Pro updates by opening the *Software Updater* tool (again, search for it in the applications menu) and entering your credentials in the Ubuntu Pro tab.

There are some additional benefits to Ubuntu Pro, most notably 10 years of support instead of the usual five. Note that support here means security updates and bugfixes, as opposed to a 24/7 helpdesk. This Extended Security and Maintenance (ESM) support applies retrospectively, so if you have Ubuntu 16.04 running somewhere, you can keep it shipshape until April 2026. In fact, if you have the previous LTS release, Ubuntu 14.04 (Trusty Tahr), running, Canonical has announced that paid ESM support is now available for this (until April 2026). Just because you can do something doesn't mean you should, though, and in general, you should upgrade to the latest LTS release well before the regular support period expires.

We love a good chart and LibreOffice makes them easy. We do miss the days when Ubuntu fitted on a CD-ROM, though.

Install the Ubuntu Advantage tools and you can keep your 16.04 installation truckin'.

Add some customisation

Make Ubuntu your own by installing the applications you need and tweaking some pertinent settings.

We've covered the basics, so now it's time to perform the final steps needed to get Ubuntu ready for action. You may want to integrate online accounts – for example, you might want to have Google Drive or Microsoft OneDrive files available in the file manager. Or receive desktop notifications for Gmail, Microsoft or other email accounts. You can even integrate with MS Exchange or Kerberos resources if you want to connect to corporate environments. All these (and more) are available in the Online Accounts section of the *Settings* app.

Earlier we mentioned installing new applications and introduced the idea of repositories, but we didn't give you the whole story. The traditional repo system meant that apps were packaged by distribution packaging teams, rather than the app developers themselves. This ensured compatibility with that distro, but also meant that programs in the repos lagged behind what you'd find if you got the source code (from the likes of GitHub or Gitlab) and compiled it yourself. Over the past few years, a couple of new technologies have emerged that make it easier for app developers to release their wares in a distro-agnostic manner, either by including all the required libraries (this is how AppImages work) or by building against standardised runtime bundles. While Flatpak (and **https://flathub.org store**) is gaining popularity for GUI apps, Canonical is pushing ahead with its own format – Snaps.

Apart from making life easier for devs and distro teams, these universal packaging formats all offer some degree of confinement for increased security. Snaps and Flatpaks have fine-grained permission management, so they can only access the resources they need. When you search for something in the App Center, the search bar separates the results into Snap packages and Debian packages (Ubuntu is based on Debian). All the featured apps there are Snaps nowadays, and Canonical is keen to point out that Snaps can be used to package system services and server tools, which Flatpak is not designed for.

Good (sn)apples and bad apples

The problem with allowing anyone to publish on the Snap Store, or otherwise put their packages in Flathub or other storefronts, is that some people will attempt to distribute malware this way. The Snap Store and Flathub both give well-known apps a verified tick, but neither has the resources to verify every contribution. Not long before we wrote this, a developer was found to have uploaded 10 'cryptocurrency wallet' apps to the Snap store *(see News, p.7)*. Not one of these was legitimate; in fact, all of them would nefariously drain
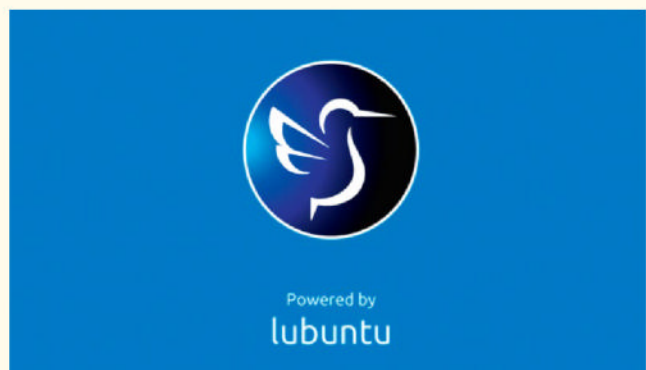
» WHAT'S YOUR FLAVOUR?

The most popular Ubuntu flavour is probably Kubuntu, which uses the KDE Plasma desktop – a fully featured, stylish affair that (by default) has a traditional taskbar and apps menu set up. The latest KDE Plasma 6.0 arrived too late to make it into Ubuntu, but we're running a desktop special showcasing it next issue. Tune in to find out how to get it on Ubuntu 24.04.

If you remember the original Ubuntu, or the Gnome 2 desktop that it was built upon, check out the MATE flavour. This is lightweight and prides itself on "traditional desktop metaphors", while providing modern conveniences (such as being able to insta-search

the applications menu). The lightest flavour is Lubuntu, which uses the LXQt desktop. Or for something in between, try the Xfce-based Xubuntu. There's also Ubuntus Cinnamon and Budgie, which use the desktops from Linux Mint and SolusOS respectively. And for Chinese users, there's Ubuntu Kylin.

If you remember Canonical's in-house Unity desktop (which reigned from 2010 to 2017), or like the look of the screenshot on the previous page, you might want to check out the Unity flavour. After Canonical ditched its desktop, a brave hacker took on the mantle and resurrected Unity with

modern application/library support. It's been an official flavour since Ubuntu 22.10. Read more about Ubuntu flavours at **https://ubuntu.com/desktop/flavours**.

Finally, besides the desktop flavours, there are

also EduBuntu and Ubuntu Studio. These both use Gnome but the former comes packed with tools for education, and the latter comes with everything you require to make music, art or video.

Lubuntu's elegant and lightweight desktop is hard to fault.
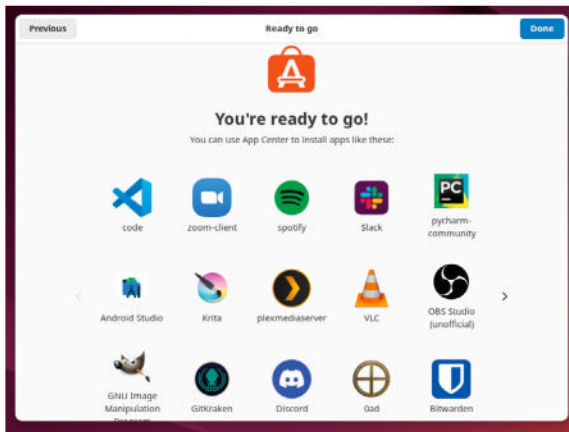
These and other exciting, entertaining and productive apps await you in the Snap Store (or App Center, or whatever).

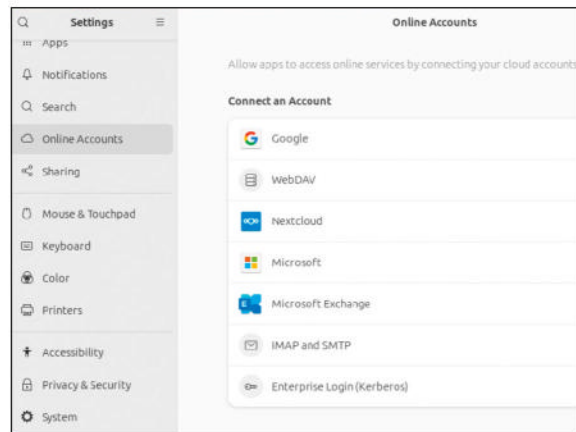You can configure everything (including Online Accounts) from the Settings panel.

any wallet a user attempted to open with them. Flathub has had similar problems, so if in doubt, stick with verified apps or Debian packages and you'll be fine.

One of the most popular Snaps (and with good reason) is the *VLC* media player. You might be familiar with it and its long-held ability to play pretty much any media file (or network stream) you throw at it. If you're into 3D rendering, you'll have heard of *Blender.* It, too, is available on the Snap Store. If you look at its listing there, you'll see it has 'classic' as opposed to 'strict' confinement. This is so it can harness the system's GPU for hardware acceleration. Classically confined apps offer the same isolation as traditional Debian packages; that is to say, none at all. So, be extra wary of unverified, unconfined apps. If you prefer working with 2D images, *Pinta* is great for drawing or bread-and-butter image manipulation. If you want something more akin to *Adobe Illustrator, Krita* is highly acclaimed.

Microsoft's *Visual Studio Code* (aka *VSCode* or *Code)* has made waves in the coding community. Say what you want about Microsoft, but there's no denying *Code* manages to do everything a traditional integrated development environment can do, while looking clean and being easy for beginners to use. The *Code* Snap is the easiest way to get the latest features on Linux, so why not give it a shot? Or maybe you'd like to have a crack at making a game. In which case you'll find game engines like the popular *Godot* or lesser-known *Unfold.*

Much of what's new in the flagship Ubuntu comes directly from Gnome 46. That release lined up nicely with Ubuntu's, so the stylish Status menu, revamped notification area and pragmatic Settings panel all feature. You might not get on with Gnome (or Ubuntu's take on it). Don't worry, you're not alone. Check out the other Ubuntu flavours, which we've summarised in the box *(opposite).* Or look at *Gnome Tweaks* in the App Center. Once installed (it's called *Tweaks* in the app menu), you can change some appearance and font settings that aren't available in the regular panel. It's worth noting that the left-hand Dock is an Ubuntu addition, so if you want to experience pure Gnome, disable it from the Ubuntu Desktop tab in *Settings.*

A few of you will want to do some gaming that's a little more serious than the *Mines* app we installed at the start. Gaming on Ubuntu, and Linux in general, has become very popular since Steam launched the Steam Play feature, which enables Windows titles to run (for

the most part) flawlessly. There's a *Steam* Snap in the App Center. In the past we'd recommend going with a traditional Debian package installation, but the Snap is in excellent standing these days. There are even candidate, beta and edge channels if you want to try the latest features. You might also want to look at the open source *Lutris* or *Heroic Game* launchers. *Lutris* works with the likes of the GoG and Humble stores (and Steam), and *Heroic* works with Epic and Amazon (and GoG). They both make it easy to choose different versions of *Wine* and *Proton* (the magic that underlies Steam Play), in case your favourite game doesn't work.

If you've read through this and are overwhelmed, or otherwise don't fancy messing with your current setup, we should point out that virtual machines (VMs) may be of use. Using a (cross-platform) tool like *VirtualBox,* you can easily set up a VM and direct it to boot the ISO image. From there you can follow the installation instructions here, and you end up with an isolated virtual system that you can meddle with risk-free. This is a great way to try out other Linux distros, too. A lot of those are based on Ubuntu (Linux Mint, Pop!_OS and KDE Neon are all popular) and will have new releases for you to try once they ingest the new version.

And with but a few moments before going to press, it's time to wrap up. Let us know how you get on with the new release, and what you think we should have warned you about. Next issue, we're looking at KDE Neon – Ubuntu with the latest KDE tooling. Also, when it happens, we'll be covering the much-hyped Ubuntu Core Desktop, an immutable (unbreakable in lay terms) OS based entirely around Snaps. We can already feel the rage from some of you, so we'll get our coats. LXF

We were pleased to see TMNF (the most popular PC racing game) was prominently advertised in the App Center.
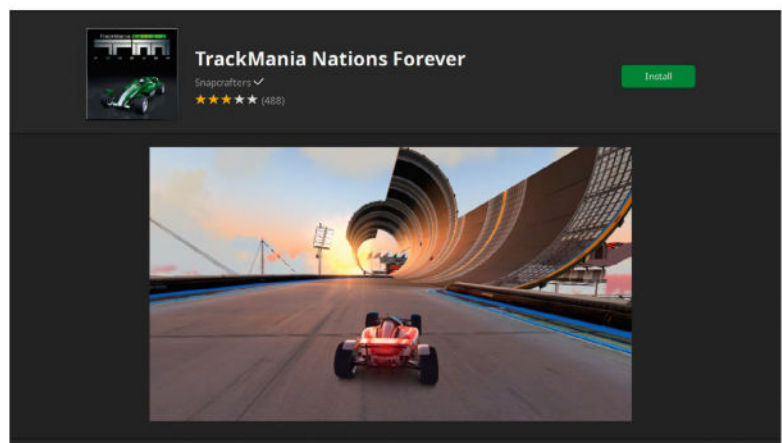
Ubuntu at 20

A decade on from celebrating a decade of Ubuntu,
Neil Mohr wonders where all the time has gone.

CREDIT: Wikimedia/Ubuntu

So, that Ubuntu logo with the three dots – it's people holding hands, from the original Warty login.

UBUNTU TIMELINE

1995	1999	2001	2004	
Debian developer Mark Shuttleworth builds the Apache web server packages for the early Debian project and is sucked into the open source community.	**Thawte sold** Shuttleworth sells his startup website certificate and security company to Verisign for $575 million.	**Foundation** The Shuttleworth Foundation is launched, a philanthropic venture to back social change by funding those with big ideas.	**Canonical formed** The UK business behind Ubuntu is founded as an independent commercial arm providing development, marketing and worldwide services.	**4.10** **Warty Warthog Kernel 2.6.8** Original release available on free CDs. Shipped with Gnome 2.8 desktop and *Firefox 0.9.*

Without Ubuntu, the current Linux landscape would be unrecognisable. Back in October 2004, the first 4.10 (2004.10) release of Ubuntu, with its intriguing Warty Warthog code name, leapt from obscurity to being one of the most downloaded Linux distributions of the year. And that's in spite of it sporting a less-than-attractive brown wallpaper. Perhaps the motto of "Linux for Human Beings" might have been on to something – radical departures such as enabling user accounts to make system-wide admin changes flew in the face of the classic novice-baffling Linux behaviour of the time.

Being backed by an actual for-profit organisation was another departure, rather than the rag-tag hacking teams or lone coders that had preceded it, with those successes coming more by accident than by design. It seemed Ubuntu was set up for success from the start. The vision came from Mark Shuttleworth, a Debian developer who benefited from the dotcom bubble, which turned him into a multi-millionaire and happy philanthropist. His passion to give back to the open source community that had helped establish him, helped establish Ubuntu – a Zulu word meaning humanity to others – a Linux distro made for humans.

He was clearly on to something, as this human-first approach created the most popular distro of all time, which went on to directly spawn more respins than anything before or since, alongside possibly more controversies than any other project has experienced, too! Fun times for all. So, after 20 years of gently shifting-hue backgrounds, let's look at how Ubuntu has developed over the decades, the controversies that exploded with that and the players behind it.

Slack and snacks

Cape Town, South Africa, early '90s. A young student is sitting late at night in the University of Cape Town's computer lab, a pile of snacks on his right and a pile of Slackware install floppies on his left. He doesn't know it yet, but he's about to become a major force in the Linux open source world; he's also going to be in big trouble, as he needs to reinstall Windows on the PC before the labs reopens, too. That student was Mark Shuttleworth, and like so many before and since, free access to open source changed his life completely.

Shuttleworth didn't have his own PC, so the only way to try Linux was on university equipment. He got involved in a project to hook up the university to the internet and started using Debian. Realising Apache wasn't available, he became a Debian developer, maintaining the first Apache package. By the mid-'90s Shuttleworth had graduated with a degree in finance

and information systems and established his company, Thawte, in the security and verification sector, built on Debian, Apache and *MySQL*. It was very successful and was bought by Verisign in 1999 for US$575 million (just over US$1 billion in today's money).

So, what's a 27-year-old multi-millionaire supposed to do with all his time and money? Other than pay to be the first South African in space… Luckily for the world, Shuttleworth has a strong philanthropic streak. He'd already established the Shuttleworth Foundation and

» SPINS, SPINS AND MORE SPINS!

Why reinvent the wheel when you can just respin it? It's testament to the success and appeal of Ubuntu just how many respins have been created over the years. According to the Linux Distribution Timeline project (**https://distroware.gitlab.io**), almost 90 distros – many now unmaintained – have been spun out from Ubuntu. Some can be as straightforward as running an alternative desktop environment, others are themed, such as Hannah Montana Linux, or dedicated to a specific task, such as Ubuntu Studio, while a number offer a completely redeveloped experience, such as Linux Mint.

There are two distinct spins of Ubuntu: there are classic Ubuntu flavour spins, then everything else. Flavours (**https://ubuntu.com/desktop/flavours**) are official spins of Ubuntu. These are supported within the Ubuntu community and backed by its infrastructure for builds and deliveries. Kubuntu was the first – this packaged the KDE desktop and associated apps for Ubuntu, rather than Gnome – and it laid the ground for how these flavours are handled, along with their relationship to Canonical and the main Ubuntu project. Other spins, such as Linux Mint, while they might take and use the Ubuntu base repositories, have to build and maintain their own releases and support services such as websites, forums and funding.

If you want a specific desktop build or have a dedicated use for Ubuntu, there's quite possibly a flavour already available.

2005
Ubuntu Foundation
Formed with a $10 million seed fund from Shuttleworth, who became the Self-Appointed Benevolent Dictator for Life, the foundation aims to ensure Ubuntu will always be available for free. It's worked so far!

2006
6.06 LTS
Dapper Drake
Kernel 2.6.15
Includes a Wi-Fi network manager, revised theme and the GUI Ubuntu Installer with merged live CD installer.

2007
7.04
Feisty Fawn
Kernel 2.6.20
Includes Windows migration assistant, restricted driver/codec install, WPA and Wi-Fi support.

2008
8.04 LTS
Hardy Heron
Kernel 2.6.24
With PulseAudio, Netbook Remix interface, *Wubi* Windows single-file install option, *Brasero* disc burner.

told us in **LXF71**, "I kind of need to get rid of everything that I've acquired. I'm quite keen to do that in my lifetime." *(You can read part of the interview on p.42.)*

As a then Debian developer and with a desire to help the open source community that had helped him, Shuttleworth devised the idea of putting a team together that really understood Debian, with a focus on bringing it to the world to benefit millions of people. This became the vision of Ubuntu. Shuttleworth had

» CONVERGENCE CATASTROPHE

You can't blame a company for dreaming big. Back in July 2013, Ubuntu announced a fresh crowdfunded phone on Indiegogo, with a target of an eye-watering $32 million in 30 days. Many thought Ubuntu wouldn't achieve this target, and it didn't, but the $12.8 million it did hit was the highest achieved by such a crowdfunded project at that point. That phone, the Ubuntu Edge, never saw the light of day, but the attention and response the project generated must make it one of the most successful failures of all time.

On the back of the publicity, Ubuntu pushed ahead with its open source phone OS. By May 2015, the first two Ubuntu Phones running Ubuntu Touch had landed. First was the BQ Aquaris E4.5 Ubuntu Edition and then the Meizu MX4 Ubuntu Edition *(see reviews* **LXF197***)*. It wasn't until February 2016 that the first tablet launched, with the Aquaris M10 Ubuntu Edition *(see review* **LXF212***)*.

Shuttleworth's original hope was to gain traction in emerging marketplaces where Ubuntu was more widely used, such as India or China. But the lack of software ecosystem meant, despite being fast, stable and having regular updates, it couldn't compete with low-cost Android, which had just released v5 on the back of the Nexus 6, and popularity king iOS, with the iPhone 6. With both devices and ecosystems backed by two of the richest corporations in history.

❗ The original BQ Ubuntu Phone gained some fans.

considered trying to steer Debian as Project Leader but he rejected that as being unworkable and instead took what he knew would be the first of many controversial choices of founding Ubuntu from several projects. But Shuttleworth wasn't afraid of controversy – a whole founding point of the Ubuntu project was the ability to steer it in the direction he wanted it to go.

On 19th October 2004, the first Ubuntu 4.10 ISO was released. You can still download the build from **http://old-releases.ubuntu.com/releases/4.10/** – there's even a build for PowerPC. Even back in 2004, it was a struggle to keep the file size to a single CD, which is less than 640MB. Shuttleworth said, "That's a tremendous narrowing. We've sacrificed a tremendous amount to do that. If you care passionately about everything that's not on that disc, we're not much help to you." Keep in mind that a decade later, the 14.04 LTS 32-bit desktop build stood at 1GB – so required a DVD – and a decade after that, the 24.04 LTS build is over 5GB, outgrowing a single-sided DVD image.

When *Linux Format* got its hands on Warty Warthog – the name came from a joke on a Sydney ferry trip about it being a rough first release – the standout was the ease and speed of installation. At the time, most distros came with a complex installer that asked you all about partitioning, then presented you with a list of packages that you had to choose from, either individually or in groups, and then spent ages installing them. Ubuntu basically asked, "Where do you want me to put it?" and then got on with it, at speed. As there were no choices to make, there wasn't the lengthy process of installing individual packages from RPM or DEB files on the disc; Ubuntu simply unpacked an archive into your root partition. This meant you could be using it less than 20 minutes after booting the CD.

For the first few releases, the live and install CDs were separate; it wasn't until PCLinuxOS showed that it was possible to combine the two into one that Ubuntu CDs became even easier to use, and you could try out the distro while it was installing.

The first bug posted for Ubuntu, called Bug #1, was from Mark Shuttleworth and stated, "Microsoft has a majority market share" **(https://bugs.launchpad.net/ubuntu/+bug/1)**, with a large part of this being that PCs were sold with Windows pre-installed. Shuttleworth closed Bug #1 in 2013, citing a much wider range of devices running Android and that cloud computing had flipped the situation on its head, forcing Microsoft to support all operating systems. But you try buying a desktop PC without Windows pre-installed…

Another controversial departure was Ubuntu's lack of software choice. The distro contained exactly one of

UBUNTU TIMELINE

2009	2010		2011	
9.10	**10.04 LTS**	**Purple reign**	**11.04**	**11.10**
Karmic Koala	**Lucid Lynx**	The new Light theme	**Natty Narwhal**	**Oneiric Ocelot**
Kernel 2.6.31	**Kernel 2.6.32**	introduced with 10.04	**Kernel 2.6.38**	**Kernel 3.0**
Ext4 becomes default,	New theme arrives,	was a stark departure	Unity becomes the	Unity 2D released,
Ubuntu One arrives	Plymouth debuts,	from the previous	default desktop,	Gnome 3.0 used,
and Ubuntu Software	Windows controls	browns to the current	includes *Banshee*	auto-hiding UI
Center is introduced.	moved, Nvidia support,	purples, alongside a	music player, *Firefox 4*,	elements, *Thunderbird 7*
USB 3.0 and Grub 2.	*GIMP* dropped.	fresh logo.	*LibreOffice, OpenStack*.	replaces *Evolution*.
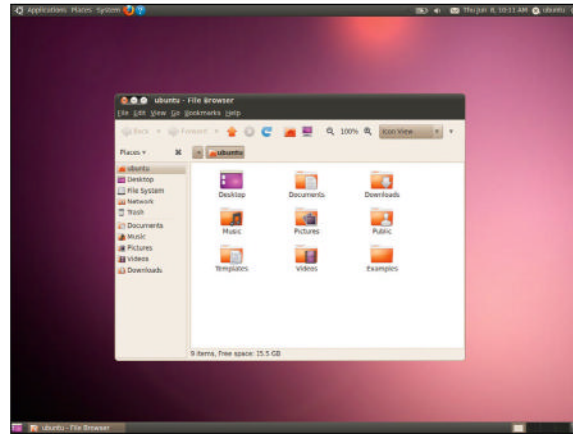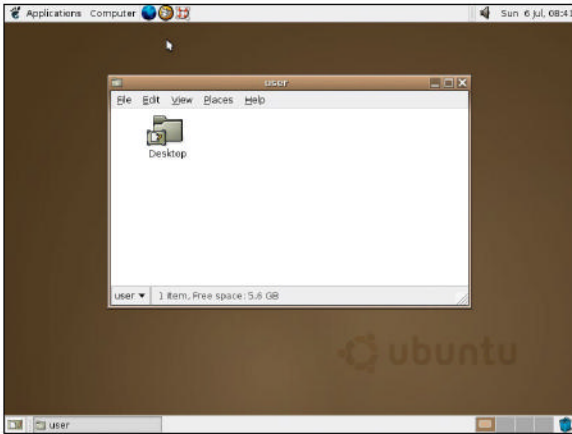
The shifting hues of the Ubuntu desktop, from Warty in 2004 to Minotaur in 2023.

CREDIT: Wikimedia/Ubuntu
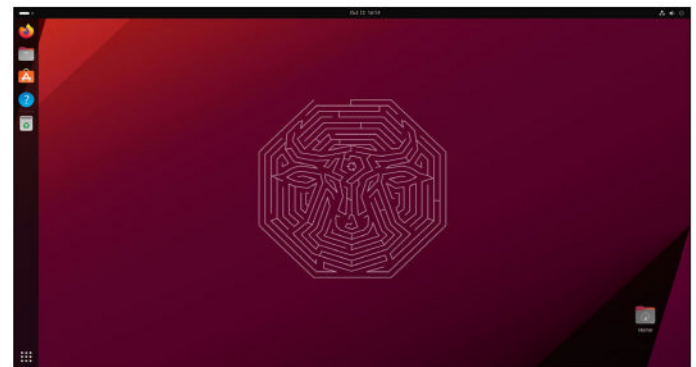
each main type of program: one desktop, one browser, one mailer and so on. The user was not offered options and were simply presented with a working system that could be changed later. Having a single set of defaults meant they could concentrate on making the initial experience as good as possible, which meant that Ubuntu quickly became the Linux for non-geeks.

Debian dissent

The first couple of releases of Ubuntu were huge hits, really causing ripples in the Linux community and one that was known for its flame wars. Soon there were a couple of major issues rumbling through the mailing lists and IRC chats of the developer communities. A number of Debian contributors were not happy with the success of Ubuntu, the way it was taking Debian sources and recompiling in a way they weren't binary compatible with Debian, and lack of compensation (or straight hiring of Debian developers) from Canonical that was being gifted to Ubuntu developers. This was reaching a peak in early 2006 but was happily diffused at the Debian (Debconf) conference in July that year with a Ubuntu/Debian powwow, the minutes of which

are still available (**https://lists.debian.org/debian-project/2006/06/msg00278.html**).

Being the popular new distro in town did mean there was a constant rumour mill swirling around anything it did. One rumour was that Canonical, being commercial, meant it was going to start charging for Ubuntu – whether that was a Pro version or Ubuntu Vista – alongside questions over trademark infringement and people's ability to create spins of Ubuntu. These were raising their heads towards the end of 2005 but Mark Shuttleworth took to the web with a comprehensive FAQ (**https://wiki.ubuntu.com/MarkShuttleworth**) outlining Canonical's commitment to free software, how spins would work – the KDE version of Ubuntu called Kubuntu had been released by this point – and explaining why Ubuntu isn't necessarily binary compatible, which is, in fact, a feature not a bug.

Going its own way

As a sign of the future, for the Ubuntu 6.10 release Canonical introduced a new init – the first process (PID1) that orchestrates service and device startup – called *Upstart*. For decades, the init was *SysVinit* on

2012

12.04 LTS
Precise Pangolin
Kernel 3.2+
Five-year LTS support, *Rhythmbox* returns, HUD arrives, DHCPv6 support, plus multi-monitor support.

12.10
Quantal Quetzal
Kernel 3.5
Unity 2D removed, Amazon arrives with its privacy controversy, Python 3, ISO bigger than a CD, *Firefox 16.*

2013

13.04
Raring Ringtail
Kernel 3.8
Wubi dropped, better privacy features, new-look exit menus, Ubuntu One Sync, UI animations.

Bug #1 closed
Shuttleworth closes the original Ubuntu Bug #1, stating that Microsoft maintains a monopoly on computer operating systems.

2014

14.04 LTS
Trusty Tahr
Kernel 3.13
Integrated menus back, Unity Control Center, TRIM, ARMv8 support, Nexus 7/10 support.

›› MARK SHUTTLEWORTH

Interview with the Ubuntu founder from LXF71, October 2005

Linux Format: *Why did you create Ubuntu?*

Mark Shuttleworth: First, a strong desire to give back. I was incredibly lucky during the dotcom boom and a lot of the reasons for that luck came from the fact that I was exposed to open source software at just the right time. I was able to build Thawte on Linux, *MySQL* and Apache. I had a very strong philanthropic programme – I kind of need to get rid of everything that I've acquired. I'm quite keen to do that in my lifetime, or in a time that would reasonably approximate what that should be. To me, this is something that I both feel great about as philanthropy and also see as a speculative investment in having at least some role to play in what the future of the software industry looks like.

LXF: *Why not steer an existing distro towards your vision?*

MS: The only one that I thought was really compelling is not steerable! In a very real sense, what we do is the product of my having thought about that. I considered standing for Debian Project Leader, but I figured that there's another way to have the same effect really, and that is to create something that really executes the vision. Make those ideals freely available, and let other people take from that what they will – take the best bits.

LXF: *Yes, it is really popular, and it seemed to come from nowhere...*

MS: Well, to contrast it with some of the other ventures out there, we just didn't compromise on some of the technical attributes. And I give the team a very free rein to run it technically the way they want to. Building on Debian obviously helps tremendously.

LXF: *Some people have said that Ubuntu has been successful at the expense of Debian. Do you think that's a fair criticism?*

MS: I certainly think it does matter that people feel that way, because I need Debian to be successful and for people to see what we do as constructive. So, it worries me whenever that gets said. At the same time, I knew when I started this that it would be controversial within the community, because there are often knee-jerk reactions. One of the reasons I again decided not to do this within Debian was because I firmly believe that there's nothing an open source team or community can't do – except do everything.

LXF: *It fits on to one disc...*

MS: That's a tremendous narrowing. We've sacrificed a tremendous amount to do that. If you care passionately about everything that's not on that disc, we're not much help.

LXF: *Do you think Linux must conquer the desktop?*

MS: Linux is absolutely ready for some desktops. So I don't really get stressed trying to convince somebody that Linux is ready for the desktop. I really shift the discussion towards figuring out which desktops they're responsible for.

Unix systems, which is effectively a series of static shell scripts. *Upstart* followed the Mac OS take called *Launchd* and dynamically managed services, networking and hotplug devices in parallel. Despite hesitation and reluctance from many, Canonical provided support and backwards compatibility with *SysVinit* and it's a credit to *Upstart* that it stayed until 2015, when the industry transitioned to *Systemd*.

The first real controversy to hit Ubuntu was the ironically named Unity desktop. First introduced in the Netbook edition of Ubuntu 10.10 (released on 10th October 2010 at 10.10am, to give a release date of 101010, or 42), Unity was spun out to all editions of Ubuntu 11.04. The previous Gnome 2 was showing its age and Gnome 3 was under development, but Ubuntu decided to go its own way.

Unity is Gnome 3 underneath, but it's a replacement for the Gnome Shell – not for the whole of Gnome 3. However, no distro had been released with Gnome 3 at that time and Unity was a shock to the system. There were three main reactions: acceptance of the new workflow, panic and a rapid exodus (Linux Mint sprang from this), and sympathy. KDE users had already suffered a similar situation moving from KDE 3.5 to 4.

Unity was the biggest cause of dissent as it had taken away the familiar and replaced it with something different – a sure-fire way to grab people's attention. It didn't help that early releases of Unity contained bugs that made the transition less acceptable for some, even though bugs are inevitable in new software. Unity gradually achieved greater acceptance and became more stable and feature-rich – until it was removed.

An honest privacy concern cropped up with the introduction of the Amazon search result integration

❚ Ubuntu has powered enterprise, the cloud and edge computing for years.

UBUNTU TIMELINE

2015

15.04
Vivid Vervet
Kernel 3.19
Systemd arrives on Ubuntu, pushing aside its own *Upstart,* improved GPU support with Radeon drivers.

2016

16.04 LTS
Xenial Xerus
Kernel 4.4
First LTS to run *Systemd,* Snaps are introduced, Python 3 is default, ZFS support, AMDGPU now default, LXD for OpenStack and Ubuntu Software.

2017

17.04
Zesty Zapus
Kernel 4.10
Driverless printing introduced, final Unity release, no 32-bit PowerPC support, swap files now default.

17.10
Artful Aardvark
Kernel 4.13
Gnome 3 desktop introduced, replacing Unity, Wayland sessions now available, 32-bit desktop builds dropped.

into the *Shopping Lens* in Ubuntu 12.10. Before this, the *Lenses* search tool enabled users to search their own documents; with 12.10, the search returned suitable Amazon results. This caused a privacy furore, especially as the feature was enabled by default. Since then, Ubuntu has anonymised data to and from Amazon, and then made this feature opt-in. There was a similar but longer ongoing issue with the Ubuntu terminal's Message Of The Day services, which seemed to be dispensing adverts – though it's more news stories. The dynamic MOTD system was introduced in Ubuntu 17.04 and regularly crops up in posts, with users perplexed by it.

Ubuntu continued to steer its own course with the next big direction change, by starting development on the Mir display server rather than stay the industry course with Wayland. This caused controversy within the Linux community, as it clearly fragmented efforts to standardise the display layer. The first iteration of Mir appeared in Ubuntu 13.10 and Canonical continued development of it until Unity 8. Ultimately, while the decision to forge ahead with Mir gave control to Canonical, the development costs were simply too high and it would eventually revert to Wayland.

More of an annoyance, the helpful Ubuntu One synchronisation and storage service that was introduced with Ubuntu 9.04 was sunset with Ubuntu 14.04. Originally it offered 5GB of cloud storage for free – this was provided on Amazon S3, to the annoyance of some – synchronised over compatible devices. Windows and Mac OS clients were provided alongside a music store and Android photo backup. Canonical did provide a grace period for users to migrate their data to alternative cloud storage services and the Ubuntu One single-sign-on service persists.

X marks the drop

The release of Ubuntu 16.04 LTS Xenial Xerus marked the beginning of the end for a host of Canonical-pushed technologies and ultimately the end of its convergence dream. Canonical's own init *Upstart* was dropped and replaced with *Systemd.* While 16.04 retained Mir and Unity 8, this was their last gasp; the following year both the Mir display server and Unity desktop would be dropped in favour of Linux industry standards Wayland and Gnome. The dream of cross-

The convergence dream never quite got off the ground.

mobile-desktop convergence went with them. Ironically, despite much flaming over the introduction of Unity, the project lives on (**https://ubuntuunity.org**), with a solid user base for its Ubuntu Unity build.

One technology that survived the cull are Snaps. Created as a way to use sandboxed and containerised software easily across platforms, Snaps were originally envisioned for servicing desktop and mobile devices as a single Snap package from the Snap Store. With initial size and speed issues – sandboxed formats tend to be far larger and the extra size plus additional service resources mean longer start-up times – alongside the proprietary nature of the format, this ground many users' gears. Canonical insisted on retaining it as a way to support its new Ubuntu Core venture and provide easier packaging for complex desktop applications, such as *Chrome* and *Firefox,* with the latest move being towards an entire Snap-based desktop Ubuntu.

From 18.04 LTS onwards, Ubuntu has entered a period of relatively stable development. Canonical has spent the last half a decade focusing on turning Ubuntu into a slick money-making business machine, trying to follow in the footsteps of Red Hat and OpenSUSE as enterprise giants and lining itself up for flotation on the stock market. It's clear that while Ubuntu was originally launched for humans, it's certainly now primarily being maintained for business. But if your main users are now cloud and enterprise, isn't that just making good human sense? **LXF**

| **2018**	**2020**	**2022**	**2024**
18.04 LTS	**20.04 LTS**	**22.04 LTS**	**24.04 LTS**
Bionic Beaver	**Focal Fossa**	**Jammy Jellyfish**	**Noble Numbat**
Kernel 4.15	**Kernel 5.4 LTS**	**Kernel 5.17**	**Kernel 6.6**
X11 returns as default, Snap default apps, Minimal install option, *LibreOffce 6.0,* emojis are now in colour!	Reboot-free kernel updates, exFAT, *WireGuard* VPN, Python 2 removed, ZFS install option, Snap-only app installs, OEM boot logo, Pi 4 support.	First LTS to support Raspberry Pi Zero to Pi 4, *Firefox* is now only a Snap, new Dock settings, Dark theme, RISC-V support.	12 years' paid support, new installer, Gnome 46 with updated Settings, revised *Software Centre, Security* app, TLS 1.0/1.1 disabled, new Snap notifications. Happy 20th birthday!
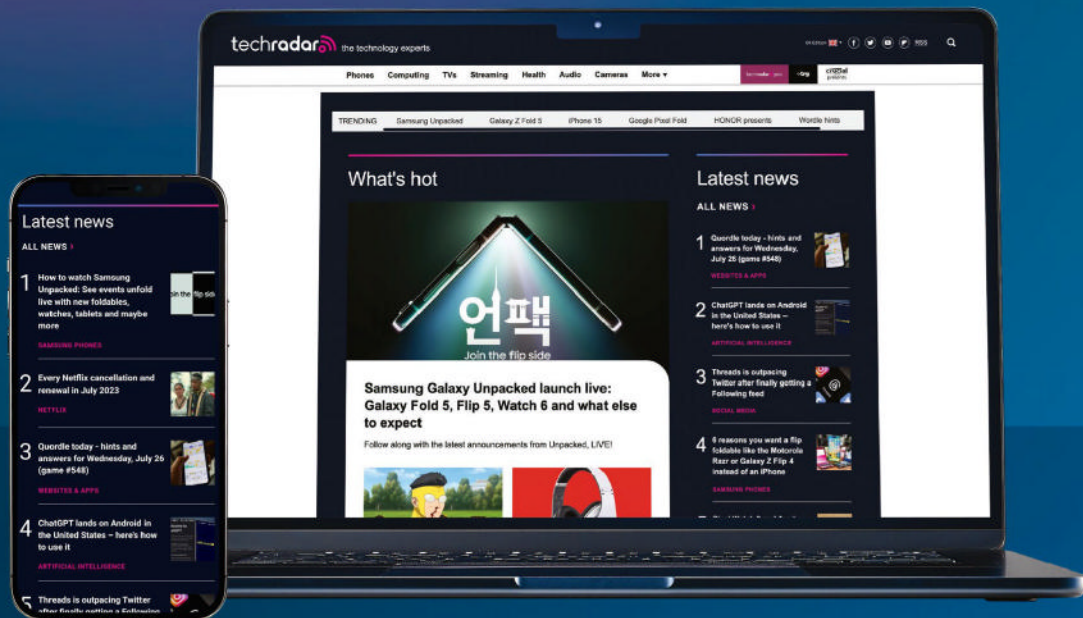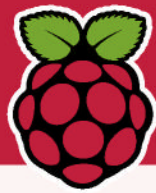
# Raspberry Pi OS 5.2 has plenty on offer

Interface speed improvements arrive alongside plenty of software updates.

The key improvement to the latest Raspberry Pi OS 5.2 release is a bump to support the most recent Linux kernel 6.6 LTS. The Debian 12-based operating system has had plenty of tweaks and comes with *Chrome 122* and *Firefox 123*. Interestingly, it also features a new version of the Pi firmware and this can be flashed using the latest *raspi-config* tool to the Pi 4 and the Pi 5's onboard EEPROM.

Improvements to the current Wayland session have been implemented and there are speed boosts for opening the system Bluetooth and Network menus. This release also improves compatibility with alternative window managers and enhances the dark theme introduced in the previous release by ensuring some widgets are now correctly displayed. Plus, the popover windows from the taskbar have been replaced with conventional windows, the shutdown

assistant has been updated to close all user processes when logging out, and the audio icon on the taskbar is now hidden by default if no audio devices are connected.

If you're not already updated, it sounds worth grabbing from **www.raspberrypi.com/software/operating-systems**.
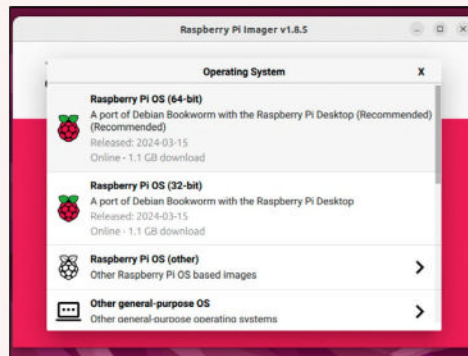
The mainstay of the Raspberry Pi world.

CREDIT: Raspberry Pi Foundation

Les Pounder works with groups such as the Raspberry Pi Foundation to help boost people's maker skills.

» PALTRY OFFICIAL Pi 5 PROGRESS

As I write this column, we are now roughly six months on from the Raspberry Pi 5's launch and things are moving forwards, but some things are seemingly not moving at all.

What we haven't seen are any official Raspberry Pi add-ons. At launch, Raspberry Pi Ltd teased a range of products. For example, an M.2 HAT for NVMe SSDs and a Power-Over-Ethernet HAT were both promised, and we saw images of these upcoming boards, but as of March 2024, we have yet to see them in the wild. We do know that they are on the way, however, because a recent listing for a European reseller advertised the M.2 HAT as coming soon.

We're also not seeing much progress on the GPIO and support for older HATs. As you may remember, the Raspberry Pi 5 has the RP1 chip, which has changed how the underlying system accesses the GPIO. No more bit-banging the GPIO; we have a proper means to work with it. However, this introduced a problem: it broke compatibility with older HATs. Work is ongoing by the board creators, but it is slow progress. The advice is that if you need to use old HATs, use a Raspberry Pi 4.

In the meantime, the Raspberry Pi community has stepped up and delivered. From the big names such as Pimoroni and 52Pi, we have seen multiple NVMe SSD add-on boards. And from new companies, such as Pineberry Pi, we have seen HAT Drive boards, which offer a range of NVMe SSD storage options and access to AI processing boards from Google (Coral TPU).

Dosbian 2.5
Here be DOS!

Written from scratch on the Bookworm OS for Raspberry Pi, Dosbian is the first distro totally dedicated to the DOS world. It boots straight to *Dosbox,* and from there, you can install whatever you want for your retro PC life. Find out more: **https://cmaiolino.wordpress.com/dosbian/**
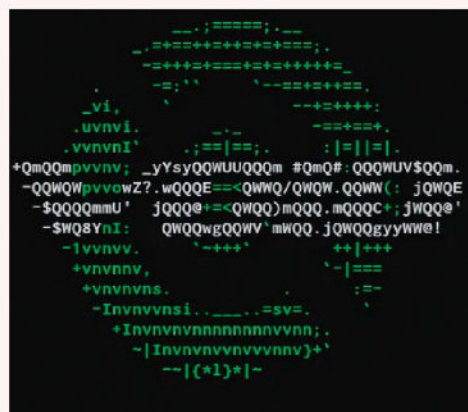

We're so retro!

CREDIT: Dosbian, Void
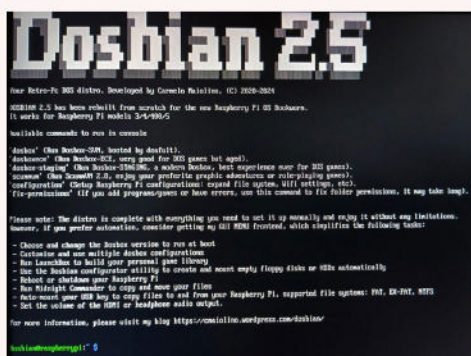
Into the Void!
ARM build out.

You can now run Void Linux, the lightweight independent distro that has a reputation for the complex, on ARM devices. It uses runit init, the X packaging system and a default command-line interface. Get builds from: **https://voidlinux.org/download/**

Diet Pi v9.1

Feeling bloated, **Les Pounder** tries another Diet, but this is no fad – he is sticking with Diet Pi this time.

IN BRIEF

A bare-bones and lightweight Debian-based OS, which is available for the first time on Raspberry Pi 5. You can build your own distro, media centre or server with very little effort. Deploy web servers, *Node-RED* or ad-blocking appliances with a useful menu system. Or you can make your own dream distro to power your projects.

Diet Pi, the low-resource Swiss army knife of distros, runs on anything, so it's fitting that there is a new alpha version for the Raspberry Pi 5.

After downloading the 170MB disk image – yeah, Diet Pi v9.1 is that small – we used *Raspberry Pi Imager* to write it to a microSD card. After the initial boot expanded the OS to use the card's full capacity, we followed the Diet Pi setup screens. This can be done at the Pi, using a monitor, keyboard and so on. Or, we can follow the setup process over SSH – look for the Diet Pi machine on your network. The process involves configuring user accounts and choosing whether or not to disable a serial console.

After setup, we are dropped into an Ncurses menu where we can further configure the system, search for software or browse the software catalogue. This is where the key strength of Diet Pi comes into play. Rather than load every single piece of software into a bloated disk image, Diet Pi provides a simple means to install what we need. In a manner similar to *tasksel* (a command used in Debian core installations), we can scroll through a catalogue of apps and set them to install. If you need a web server (LAMP, LLEP and so on), select and install – all of the dependencies are taken care of. The sheer number of installable applications and tools available via the catalogue is overwhelming, but luckily Diet Pi has broken them down into sections. From desktop environments to media/home automation servers and gaming, it is all there and so, so easy to install.

For our test, we installed *Node-RED,* Mosquitto MQTT server and a web front-end for Diet Pi. We just selected all the apps from the catalogue, hit Confirm, and then selected Install. In a few minutes we had everything running and working as we expected. All of the services were enabled and started when the Raspberry Pi 5 was booted. We soon had a *Node-RED* flow running to send and receive messages on our own private MQTT server. Handy for those of us who like building Pi sensor networks but don't want the world to see our data.

Diet Pi has an Ncurses config menu that offers tools to configure audio, networking and performance. Yes, we
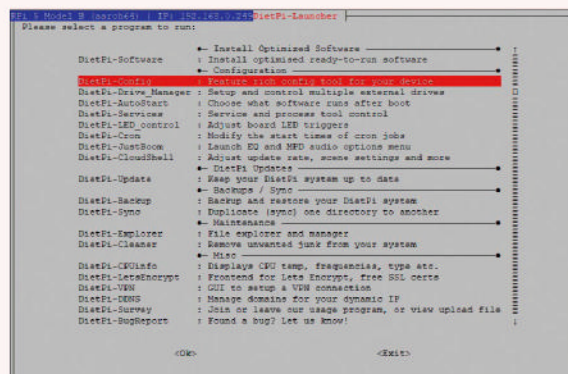
The Ncurses menu system may look a bit dated, but it can really help you to get things done – quickly.

can overclock the Raspberry Pi via this handy menu. The display menu provides a means to manage the screen resolution, but also options to add VC1 and MPEG2 keys for hardware video decoding, useful should you wish to install *Kodi* via the software catalogue. Diet Pi config has a great tool to stress test and benchmark your Raspberry Pi – handy for a Pi-focused journalist.

Why do we need a low-resource, highly customisable OS for the ultra-powerful (in terms of single-board computers) Raspberry Pi 5? The truth is, we don't, but then again, Linux users always want to squeeze every last drop of power from any age of hardware. Diet Pi is more than just a low-resource OS. The sheer amount of flexibility provided via the Diet Pi tools means we can deploy our own custom operating system for a project.

We love Diet Pi; it's powering a website on an 'old' Raspberry Pi 4 for us. This alpha release for the Pi 5 is a great start for the flagship Pi and it performs remarkably well. It builds upon the great work in previous releases and ensures that Diet Pi will be a great choice for your future projects. The Ncurses menu system is intuitive and easy to use. The software catalogue is a refreshing change from the usual kitchen-sink approach. We'll be using Diet Pi for a few more projects with our Pi 5. **LXF**
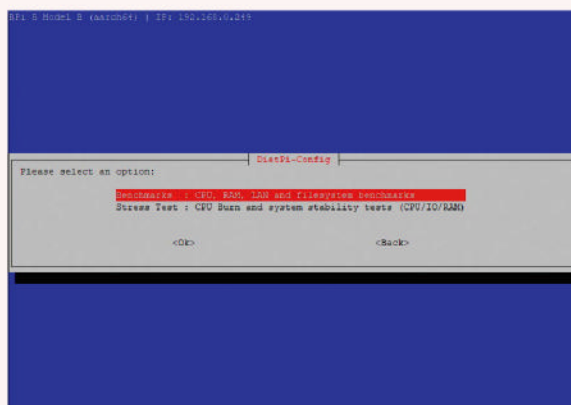
Hidden in the config menu is a benchmarking suite to test your Raspberry Pi. You can stress the CPU, RAM and IO.

VERDICT

DEVELOPER: Daniel Knight
WEB: https://dietpi.com
LICENCE: GPL v2

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

A small install size, ease of use and great selection of software make Diet Pi the ideal choice for project makers.

» **Rating 9/10**

Original Prusa XL

The race for colour is on and **Denise Bertacchi** with her colourful personality is certainly leading the pack on testing.

SPECS

Build: 360x360 x360mm
Type: Direct drive (PLA, PETG, ASA, ABS)
Levelling: Full auto
Nozzle: 0.4mm Nextruder, 290°C
Bed: PEI spring steel flex plate, 16 heated segments
Sensors: Runout, loadcell, power panic, fan speed, four thermistors
Comms: USB, LAN, Wi-Fi
Control: 3.5-inch colour with knob
Size: 800x 800x900mm, 25.3kg

The wait for a bigger 3D printer from Prusa Research has ended, and it certainly over-delivered. Instead of simply upsizing its classic i3 design, Prusa chose to create a completely new Core XY tool changer with a 360x360 mm build plate. The machine can be ordered with one, two or five toolheads.

The XL solves one of multicolour printings' most annoying problem: wasted filament and AMS jams. There is very little waste when each spool has its own dedicated toolhead, and the filament path is as simple as a single-colour XL. Not having to pull the filament in and out of a single hotend is also quicker – we saved about six hours of print time on a four-colour print, despite the XL using a slower print speed and acceleration.

It's a beefy boy with Bowden tubes that arch over the top and spool holders that mount on both sides. The frame is C-shaped, keeping the front open and easily accessible to reach in and grab your prints. The key feature is the tool changer, which gives the XL the ability to swap between materials without wasting time or filament, eliminating the need to wind material back on to the spool between colour swaps. Each spool gets its own filament path from spool holder to nozzle.

Heads up

The toolheads are sold as upgrades if you need to spread the cost out over time. One toolhead makes the XL an oversized MK4 capable of printing helmets and other large pieces. Using two toolheads creates a dual-colour beast with better precision than any IDEX we've tested. The full five beat anything from Bambu Labs in both size and number of available spool slots.

Assembly 'only' needed the toolheads to be mounted. This is still quite a process, and installing five toolheads and calibrating each of them takes several hours. Bed levelling is a no-brainer as the XL is fully automatic.

Loading filament is more complex than on the single-colour Prusa MK4. Each toolhead has its own spool holder and more than a metre of Bowden tube. The printer is direct drive, meaning you need to hand-feed
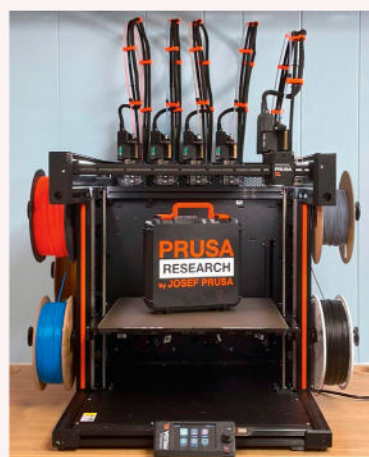
We're going to need to print a bigger boat...

the filament through a lot of tubing before the extruder gears can take over. TPU is slightly more annoying to load, as it is too soft to enter the extruder on its own.

The Original Prusa XL currently works best with its own slicer, *PrusaSlicer.* To use the XL's Wi-Fi you have to use a separate program, either the basic *Prusa Link* or the cloud-based *Prusa Connect* for full remote control.

We were curious how the XL would do with a Speed Benchy, as Prusa has no information on its top speed or acceleration rate. The slicer only shows a top speed of 200mm/s for infill, and max acceleration of 5,000mm/s^2.

The Prusa XL won't top our list of fastest 3D printers with its 25-minute and 14-second Benchy. To test its efficiency with multicolour printing, we printed a dragon using four colours. It took 13 hours and 51 minutes using default speeds of 170mm/s on the walls and a 0.2 mm layer height. The same print took eight hours longer on a faster Bambu Lab P1P and wasted 70 grams of filament.

Because of its tool changer, the Original XL is a dream come true for 3D printing fans. It's is easy to use, with hands-free auto bed levelling and the ability to set its own Z height. It's backed by Joseph Prusa's commitment to open source, service and data security, too. **LXF**

Pushing speed (seen in the right-hand print) doesn't do anything for the XL's quality.

VERDICT

DEVELOPER: Prusa
WEB: www.prusa3d.com
PRICE: £1,799 one head (£3,390 five heads)

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

One of the best 3D printers we've tested. It's a pricey multi-material tool changer that emphasises quality over speed.

» **Rating 9/10**

SCOPPY

Build your own Pi Pico based oscilloscope

Discover how to build an oscilloscope with a Raspberry Pi Pico, an old Android device and a spare **Les Pounder**.

OUR EXPERT

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

Oscilloscopes are expensive and largely reserved for makers or electronics engineers who need to understand what a circuit is doing. We can't make our own, simple oscilloscope and logic analyser with a £4 Raspberry Pi Pico, right? It turns out that with a Raspberry Pi Pico, some software called *Scoppy* and an old Android phone, we can do just that.

For the low, low price, we can't expect the earth. This won't rival an expensive setup, but for most of us, it is enough to understand and get to grips with the equipment before we open up our wallet.

In this tutorial, we learn how to configure a Raspberry Pi Pico as an oscilloscope and use the *Scoppy* Android app to read live data from a circuit. Note that due to the Raspberry Pi Pico's 3.3V logic, we cannot measure voltages any higher, otherwise we risk damaging the Raspberry Pi Pico's ADC pins.

YOU NEED

> 2x Pi Pico or Pico W
> Android or Apple phone/tablet
> 2x half-size breadboards or one large
> A 100 Ohm resistor (brown-black-brown-gold)
> Handful of M2M jumper wires
> Code: https://github.com/lesp/LXF315-Scoppy/archive/refs/heads/main.zip

Setting up Scoppy

While holding the BOOTSEL button, connect your Raspberry Pi Pico to your computer. Go to **https://oscilloscope.fhdm.xyz/wiki/Installation-&-Getting-Started** and download the *Scoppy* firmware for your Pico or Pico W. Open your file manager, go to the downloaded file and copy it to the root of the **RPI-RP2** drive. This flashes the new firmware to the Pico. When done, remove the Pico from your computer and connect to your Android device. Depending on your phone, you need an adaptor to connect. The Raspberry Pi Pico uses a micro USB port, but our old phone uses USB type C. We're using a micro-USB-male-to-USB-A lead, so a USB-A-to-USB-C adaptor is a cheap and easy way to connect the USB A side of the lead to the USB C port of the phone.

We now need to install *Scoppy* on our Android device. Open the Google Play Store on your device and search for *Scoppy - Oscilloscope.* Alternatively, you can open the store in your desktop web browser and manage the installation remotely by going to **https://play.google.com/store/apps/details?id=xyz.fhdm.scoppy**.

Open the *Scoppy* application and you are prompted to allow *Scoppy* to access Pico. This is the default name given to the Raspberry Pi Pico by the *Scoppy*
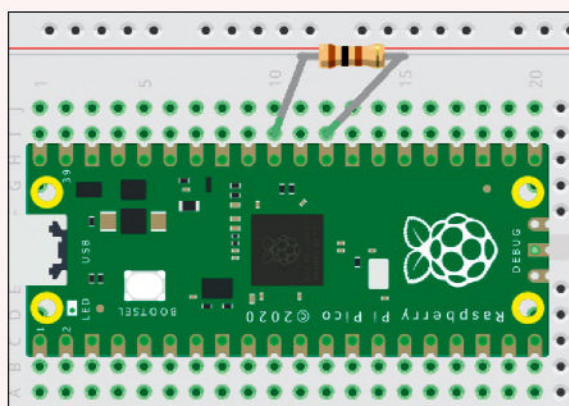
Scoppy has a test circuit that just needs a 100 Ohm resistor between GPIO 22 and GPIO 26.

firmware. Allow access and we are dropped into the *Scoppy* oscilloscope user interface. Initially this is quite an overwhelming experience.

Before we get into the details, let's confirm that our setup is working, and luckily *Scoppy* has a test signal that we can use to test it. Using a 100 Ohm resistor (brown-black-brown-gold), connect GPIO 22 to GPIO 26. GPIO 22 is running a 1kHz square wave with a duty cycle of 50%, and this will generate a waveform on the *Scoppy* display, proving that our installation is working as it should.

Using Scoppy

The *Scoppy* oscilloscope screen can be a little mind-boggling, so let's break it down and learn how it works.

At the top of the screen is the status bar and this is where we can set the sample rate (to capture a waveform), the sample count (how many samples the app stores), the sample record view (we can scroll through the entire waveform) and the horizontal position (where we are in the waveform).

On the right-hand side, we have control panels to toggle run modes (to capture a streaming waveform, or to capture a single slice of the waveform). There are horizontal navigation controls to zoom into (decrease the time between divisions) and move around the waveform. Then there's vertical navigation to increase or decrease the volts per division and to move the

waveform vertically on the screen. The bottom status bar is where we can find the signal source (our USB-connected Pico) and the channel badges. We have two channels, and the badges show us the details for each channel (volts/divisions, position and voltage range).

In the centre of the screen is the waveform view and here we can see the waveform generated by the tested circuit.

Shake that oscilloscope

We have a running oscilloscope, but we don't have anything to test. For this part of the build, we need a second Raspberry Pi Pico, running MicroPython.

While holding the BOOTSEL button, connect your Raspberry Pi Pico to your computer. Go to **https://bit. ly/lxfupdate** and download the version of MicroPython for your Pico or Pico W. Open your file manager, go to the downloaded file and copy it to the root of the **RPI-RP2** drive. This flashes the new firmware to the Pico.

Using your distribution's package manager, install *Thonny.* For the latest Ubuntu release we have to use a Snap package:

```
$ sudo snap install thonny
```

Open *Thonny* and let's write some code to use PWM (pulse width Modulation) to create a waveform for our *Scoppy* Pico to read.

We first import from Machine the **Pin** and **PWM** classes; these enable us to control the GPIO pins.

```
from machine import Pin, PWM
from time import sleep
```

Create an object, **pwm**, and in there create a PWM object that references GPIO 2. We'll use this to control the pin, to generate a waveform that we can see in *Scoppy.* We then use the **pwm** object to set the frequency to 500Hz.

```
pwm = PWM(Pin(2))
pwm.freq(500)
```

Create a **while True** loop, because we want this code to run continuously:

```
while True:
```

Inside the loop are two **for** loops. The first changes the duty cycle (how long the pin is pulled high versus low) to create a square wave that has an expanding duty cycle. The sleep of 0.0001 seconds creates a pleasant and easy-to-view waveform:

```
    for duty in range(65025):
        pwm.duty_u16(duty)
        sleep(0.0001)
```

A second **for** loop, inside the main loop but not the previous **for** loop, works in reverse. It shrinks the duty cycle and compresses the waveform:

```
    for duty in range(65025, 0, -1):
        pwm.duty_u16(duty)
        sleep(0.0001)
```

Save the code to the Pico as **waves.py** and click Run. Now, using two male-to-male jumper wires, first connect any GND connection on the Pico to the Pico running *Scoppy.* Then use another wire to connect GPIO 2 from the Pico running the MicroPython code to GPIO 26 via a 100 Ohm resistor (brown-black-brown-gold). See the diagram in the download for clarity. On
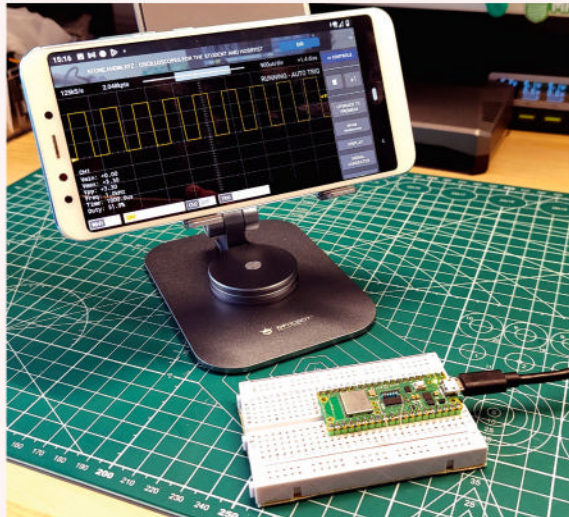
This low-cost oscilloscope setup cost less than £10 and we get to reuse an old Android device.

the *Scoppy* screen we should now see a square waveform that expands and contracts.

Logic analyser

The logic analyser part of *Scoppy* is a bit hidden. Press the Menu button, then look for Mode. Mode is initially set to Oscilloscope, but a quick tap changes this to Logic Analyzer. GPIO pins 6 to 13 are used to read multiple inputs at once (you need to make a payment of £2.79 for this) and can be used to debug circuits.

To debug a circuit you need to ensure that there is common GND between the circuit you are testing and the *Scoppy* Pico. Then connect the outputs of the test circuit to the logic analyser input pins. Remember that the *Scoppy* Pico can only accept voltages up to 3.3V.

For very little money and effort, we have a basic oscilloscope and logic analyser. We can use it to learn how to use the tools before raiding our piggy bank. **LXF**

>> GOING WIRELESS!

If you've got an £8 Raspberry Pi Pico W, you can flash the Pico W firmware and follow a few extra steps to wirelessly connect your Android device to a *Scoppy* Pico.

First, connect the *Scoppy* Pico to a 5V power source and wait for the green LED to flash four times, then pause. This means the *Scoppy* Pico is ready to connect to. On your Android device, go to the Wi-Fi settings and look for an access point that begins with SCOPPY. Connect and acknowledge that you are OK with connecting to a non-internet Wi-Fi connection.

Go into the *Scoppy* app and in the bottom-left, tap Connection and then Change Connection. Select Wi-Fi and *Scoppy* should auto-connect. You see `Wi-Fi: OK` when it is successful. The *Scoppy* Pico and *Scoppy* app should auto-connect each time they're powered up.

The wireless connection frees up your Android device to be placed in a more convenient location, and it stops the *Scoppy* Pico draining your battery too quickly (that is our one issue with the project).

For £3, *Scoppy Premium* offers more channels (two oscilloscope channels and eight for logic analyser), and this along with Wi-Fi connectivity is a great way to reuse an old Android device and a spare Raspberry Pi Pico W.

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

PYRADIO

Live-stream audio

Shashank Sharma often finds himself with song lyrics stuck in his head. After all, no one ever got in trouble for going gaga for radio.

OUR EXPERT

Shashank Sharma is a trial lawyer in New Delhi. He's been writing about open source software for 20 years and lawyering for 10.

A s opposed to music players, which can be used to enjoy your own music collection, radio exposes you to new and, sometimes, old wonders that you otherwise might not come across. While some graphical music players let you stream internet music, CLI warriors might occasionally want to move to the music, too. Thankfully, the Curses-based internet radio player *PyRadio* fits the bill perfectly.

We interrupt this broadcast with a special shout-out to our beloved editor Neil. Refer to the box *(opposite page)* to learn how one email changed the course of this tutorial. But more importantly, tune in next month to read what would have been this issue's tutorial. We return you now to the regular programming, so you can learn how to play internet radio stations from the comfort of your terminal.

A quick internet search could land you dozens of websites that let you tap into internet radio stations. You could just pick a genre – jazz or rock, say – and start listening. The downside of most such sites is pop-up ads and cookies that don't respect your privacy.

In *PyRadio,* you have a fully-featured utility that's entirely controllable with keyboard shortcuts, boasts of a neat interface, isn't a drag on your system resources, and keeps you safe from unwanted cookies and annoying ads. Even better, the utility boasts *Vi*-like keybindings, supports themes, lets you search for stations, and more.

Destination, determination, deliberation!

You'll find the three Ds of apparition as discussed in the ever popular Harry Potter books to be a useful guide

You can play almost any kind of streaming content with PyRadio, such as true crime podcasts or news.

With mouse support enabled, you can double-click to play a station, or use Shift+scroll wheel to increase or decrease the playback volume.

when installing *PyRadio.* Only a handful of distros carry the MIT-licensed Python project in their software repositories. Normally, this wouldn't be a problem, as you could use the *Pip* package manager to quickly install it. However, this is no longer possible on modern distros, such as Ubuntu 23.04 and newer, which have switched to installing Python packages in a virtual environment using the *Pipx* utility.

The installation is quite involved, requiring many dependencies such as python3-netifaces, python3-psutil, python3-dnspython, python3-dateutil, python3-requests, libpython3-all-dev and more. Thankfully, you can use the software repositories on your distro to install these. You'll have to refer to *PyRadio's* GitHub page, which provides a list of dependencies for Debian/Ubuntu and Fedora and derivative distros (**https://tinyurl.com/pyradio-install-instructions**).

With all the dependencies installed, you can install *Pipx* with the `python3 -m pip install pipx` command, followed by `python3 -m pipx ensurepath`.

With all that done, it's time to download the official install script with:

```
$ wget https://raw.githubusercontent.com/coderholic/
pyradio/master/pyradio/install.py
```

You can then install *PyRadio* using the downloaded install script with the `python3 install.py -i` command.

As of now, the quickest way to install *PyRadio* is with the Snap package. Run the `sudo snap install pyradio` command if your system is already configured

for Snap packages. Unfortunately, this yields an older release. Unlike many text-based tools, *PyRadio* is under constant development. So much so that the project checks for any available updates once every 10 days and asks you to upgrade when one is available. Also, the project has made significant changes in the latest 0.9.3 release. Although slightly more time-consuming than using the Snap, we recommend you install *PyRadio* yourself to get the latest version.

Hit play

Run the `pyradio` command to launch the radio player. The app greets you with a list of preset radio stations. You can use the up/down arrow keys, or the *Vim* navigation keys `j` and `k` to scroll through the list. Press left/right, `h` / `l` or hit Enter to start playing the selected station. The track, album and artist name are displayed in the status bar at the bottom, if available.

You can call up the help menu from within *PyRadio* by pressing `?`. The *PyRadio* Help window is scrollable, and split into various sections such as General Help, Volume Management, Searching, Recording and so on. Here's a list of commonly used keybindings, but be sure to read through the help for a complete list.

Keybinding	Function
?	Open help screen
Ctrl+n	Play next station
Ctrl+p	Play previous station
Space	Start/stop playing selected station
Esc or q	Quit
- or ,	Decrease volume
+ or .	Increase volume
t	Change theme
c	Open configuration dialog
a	Add new station
x	Delete selected station

If you dislike *PyRadio's* default dark theme, you can switch the theme by pressing `t`. Scroll through the available themes and press Space to see it in action. Keep scrolling until you find the one that catches your fancy, and now press Enter. The current theme features a * before its name in the theme selector window. When you press Enter after selecting a theme, you'll notice the * next to the name of your selected theme. Press Esc to return to *PyRadio.*

To add a new station to the list, press `a` to open the Station Editor dialog box. You're prompted to provide a name for the new station and the URL to the PLS playlist. Press the Tab key to cycle through the different fields in the Station Editor dialog and select OK and press Enter to complete the process. The new station is now featured in the list.

The latest release of *PyRadio, 0.9.3* is a major upgrade, and the developer expects and warns of bugs. In fact, version 0.9.3.1 was published to fix user-

reported bugs within a day. In our tests, we found that pressing `1` results in the Help window popping up. This makes it impossible to add a station that has a 1 in the URL. Thankfully, there's a quick workaround.

PyRadio stores the playlist of stations in the **stations.csv** file under the **~/.config/pyradio** directory. You can edit this file in your favourite text editor. The format is straightforward, with the title of the station, followed by a comma and then the URL to the PLS or M3U file. You can create different CSV files to store different stations. For instance, you could have a CSV file for your favourite music stations, another for news, and yet another for podcasts.

Configuration

You can also tweak a lot of settings, such as the default timeout value, or toggle mouse support from within *PyRadio* itself. Press `c` to open the configuration dialog box. All configuration options are listed on the left. As you scroll through the list, you can read a description for each on the right-hand pane of the configuration dialog box.

To change any config option, first select it in the list, and then press Space or Enter. If the option is a toggle between true/false, such as with Enable Mouse Support, pressing Enter or Space toggles the option, otherwise a further dialog box pops open with options for you to choose from. To see this in action, select Theme from this list and hit Enter.

If you make any changes to the config, you have to press `s` to save the change, and then `q` to return to *PyRadio's* main screen. ◪

≫ RADION

We've had an eye on *PyRadio* for a long time now but kept pushing it down the road for want of another versatile radio player. Then, one day, not too long ago, Neil sent an email recommending *Radion* as a possible candidate for this tutorial series, and we realised we had enough CLI radio players after all.

While *PyRadio* can use either *Mpv, Mplayer* or *VLC* as the media player, *Radion* requires *Mpv* to be installed on the system. You also need to install *Sox* if you wish to take advantage of *Radion's* ability to record playback, a feature that's also available on *PyRadio* but couldn't be covered due to space constraints.

Execute the following commands in order to install *Radion:*

```
$ git clone https://gitlab.com/christosangel/radion
$ cd radion/
$ chmod +x install.sh
$ mkdir ~/.config/mpv/scripts
$ ./install.sh install
```

You have to create the **~/.config/mpv/scripts directory**, if it doesn't exist already on your system, otherwise the *Radion* installation fails. You can now launch *Radion* with the following command: `radion.sh`.

Unlike *PyRadio,* which presents a straightforward list of stations to choose from, *Radion* instead offers a list of genres such as Disco, Pop, Rock, Classical and so on. After you select a genre, *Radion* presents a list of stations for that genre. Select the station from the list and hit Enter to begin playback.

≫ **ENHANCE YOUR TERMINAL-FU** Subscribe now at **http://bit.ly/LinuxFormat**

LINUX BASICS

Enhance the Gnome desktop

Fond of standing in the garden wearing a pointy hat, **Nick Peers** is the perfect person to explain how to set up and use the Gnome desktop.

OUR EXPERT

Nick Peers has had his foot in all three operating system camps for far longer than he cares to remember. But only one OS brings him joy on a daily basis – his Debian-powered server.

In recent months, interest in Linux has exploded. Increasing numbers of users are finally turning their backs on the likes of Microsoft and Apple to dip their toes into the wonderful world of open source, privacy-first computing. But even if Linux becomes ever more user-friendly as the years pass, it can still be a bit of a culture shock when you scratch the surface.

That's where this new series comes in. We're going to provide you with a beginner's guide to key parts of Linux – perfect for newcomers, switchers and anyone who wants to brush up on their core knowledge. We're going to focus on the most popular flavour of Linux out there: Ubuntu. We'll be basing it on the latest version – 24.04 LTS – and that means in this initial piece, we're focusing on Ubuntu's default desktop, Gnome.

Take a tour

When you first boot into Ubuntu, you're prompted to log into your user account. Type your password and hit Enter to find yourself at the desktop. The Gnome desktop employed by Ubuntu will feel at once both superficially similar but also quite different. For those switching from Windows, the key elements are all there, but jumbled up – it won't take long to orient yourself, however.

The annotation *(opposite)* reveals the key elements to look for. The Dash on the left provides shortcuts to key apps – *Firefox* (Ubuntu's default web browser), the *Files* app (the equivalent of File Explorer in Windows or Finder on the Mac), the App Store for quickly finding and installing new apps, and the *Help* tool.

The Dash also displays icons for any external drives (including DVD), the Trash and any other running programs. Open apps are highlighted with one or more red dots next to their icon (one dot per open window), and if the app is in focus, its icon is highlighted, too.

Once opened, favourite apps can be pinned to the Dash. Just open the app, then right-click its Dash icon and choose Pin To Dash (you can remove pinned apps – including default ones – by choosing Unpin instead).

The right-click menu is also context-sensitive, tailored to each app – for example, *Firefox* provides options to open new browser windows (both regular and private). If the app has multiple windows open,
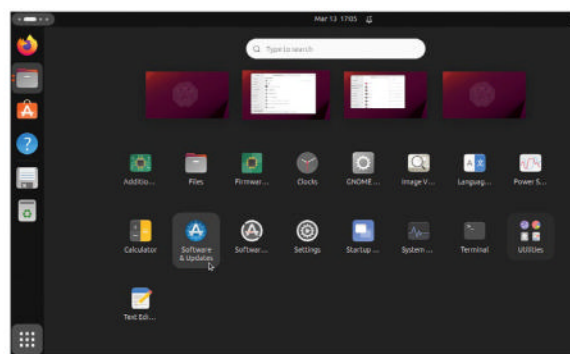
As you add new apps to your Linux installation, the Launcher plays a key role in making them accessible.

choosing All Windows displays thumbnail previews of each – click one to bring it into focus.

View app list

Apps that aren't already pinned to the Dash can be found by clicking the Launcher shortcut in the bottom-left corner. This displays the apps across one or more pages, in alphabetical order and accompanied by their icon to aid visibility. Above these you'll see thumbnail previews of all open workspaces – think of each one as its own separate desktop, enabling you to spread apps across multiple workspaces to reduce clutter, organised however you see fit. The step-by-step guide *(page 55)* reveals how workspaces work in more detail.

You'll also see a search box at the top of the Launcher – this is more than a simple app filter; it's a general-purpose search tool. Start typing and you'll see a list of matching apps followed by a series of results, split into categories: files, settings, world clocks and Ubuntu's *Characters* tool. The latter is a great way to track down Unicode symbols – simply click a result to copy it to the clipboard for pasting elsewhere.

Working in apps

Open apps operate in their own self-contained windows, as in Windows. There may be a menu bar, or the app may display key tools along its top bar, with a hamburger icon providing access to a drop-down menu with additional options. You'll see the usual controls for maximising, minimising and closing windows, while you

can move and resize app windows manually in the way you're used to – click and drag on the top bar to move it, or click and drag on an edge (or corner) to resize it.

The Gnome desktop also supports window tiling – you can drag a window to the top, bottom or sides of the screen to tile it horizontally or vertically, or drag it into a corner to resize it in quarters. Also experiment with the tiling keyboard shortcuts: Windows/Apple/Cmd key plus arrows keys.

Navigate files and networks

Open the *Files* app and you'll immediately see a familiar-looking two-paned window. As with File Explorer and Finder, the left-hand pane contains shortcuts to key parts of your system. These cover your personal user folders. Each user in Linux is given their own **Home** folder, inside which all personal files are stored in a selection of self-explanatory subfolders (**Documents**, **Pictures** and so on). It's also where many apps store your personal configuration data, too.

Should you need to venture outside this folder for any reason – to connect to external or secondary hard drives, for example – click the + Other Locations link. You can access the root of your hard drive via the link under On This Device (typically called Ubuntu), but unless you know what you're looking for, we recommend leaving it untouched for now.

Other drives should show up here too – Ubuntu can read and write to drives formatted in Windows (FAT32, exFAT or NTFS), while those wanting access to data on Mac-formatted drives should check out the Quick Tip *(over the page)*. Double-click a drive to mount it and access its contents – note that when you reboot Ubuntu, the drive needs to be mounted manually again.

Beneath this is a Networks section for connecting to shared folders over your network. This is probably empty because Linux's default file-sharing protocol (Network File System, or NFS) isn't used by other operating systems. The most common protocol – standard on Windows, and compatible with Mac OS –

THE GNOME DESKTOP UNCOVERED

1 Activities overview
New to Ubuntu 24.04, this button enables you to quickly launch apps, switch windows and manage workspaces.

2 Dash
Gnome's equivalent of the Windows taskbar and Mac OS dock. Shows both pinned and running applications.

3 Show Apps
Opens the Launcher, with shortcuts to installed apps, desktops and a search tool.

4 Notifications
Click to reveal a pop-up calendar with your upcoming events. Notifications are also displayed here.

5 System menu
Click here to reveal useful shortcuts, including access to power and lock options.

6 Workspace
The currently selected workspace, which contains any windows opened by apps while in that workspace.

is Samba, which you need to install separately. To do this, you need to install an additional package, which requires you to access Linux's underlying shell *(see the box, over the page)*. Open the terminal and then type the following two commands; press Enter after each:

```
$ sudo apt update
$ sudo apt install samba
```

You're shown a list of required packages totalling around 80MB. Press Y and they're automatically downloaded and installed. Once complete, close the window and reopen *Files,* then click Other Locations

» CHANGE YOUR DESKTOP

If you find yourself unable to cope with Ubuntu's default Gnome desktop, what other options do you have? You might want to consider switching to a completely different flavour of Linux – if you're switching from Mac OS, for example, Elementary OS (**https://elementary.io**), as featured in **LXF310**'s *Learn Linux* feature, should appeal.

However, you may be able to avoid a time-consuming OS reinstall. Linux enables you to install more than one desktop at once and swap between them from the login screen. Windows switchers looking for a more familiar desktop should consider the Cinnamon desktop instead of Gnome. To install this, open a terminal window and issue the following commands:

```
$ sudo update
$ sudo apt install cinnamon-desktop-environment
```

Reboot your system, then click your username and look for the Settings button on the login screen. Click this and choose Cinnamon from the pop-up menu before entering your password and logging in to your new desktop.

You're presented with a desktop like that in Windows 7 (and to a lesser extent Windows 10 and 11). There's a Start menu
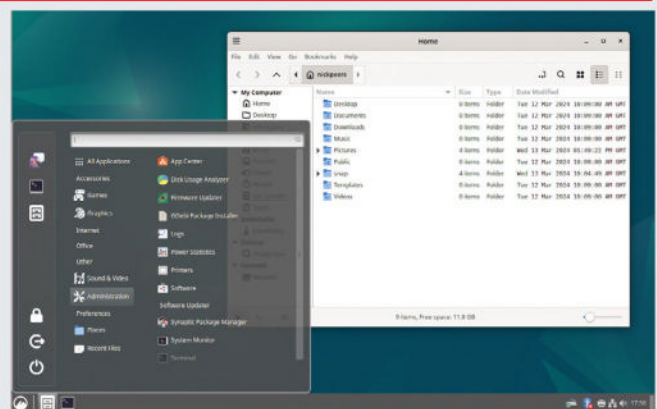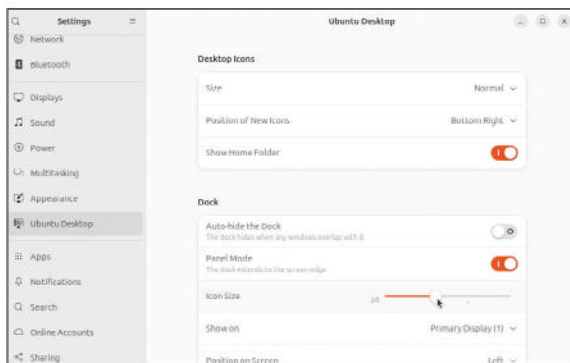

Cinnamon offers an experience closer to what Windows switchers are used to.

replacement in the bottom-left, and a taskbar that works in a similar way to the one you find in Windows. Cinnamon also ships with its own alternative to Gnome's *Files* app, which is both similar yet more configurable at the same time.

You'll find plenty of options for fine-tuning the desktop's look and feel in the Settings app. Some are window dressing, but many aren't.

again. You should now be able to see (and access) most shared folders from other devices on your network. Double-click one to see a list of all folders it's sharing, and double-click one to open it – you're prompted to connect anonymously (if guest access is enabled) or as a registered user, in which case enter the username and password assigned to that shared folder. By default, the password is remembered until you log off or shut down – select Remember Forever to save having to enter it each time you connect in future.

Search for content

Files ships with two search tools. The first can be found to the right of the address bar. This allows you to search the current folder for content. A second tool is accessed by clicking the magnifying glass, top-left.

This 'search everywhere' tool works in a similar way to Windows, which means it's not strictly speaking everywhere. Instead, it covers all your personal folders – in other words, **Documents**, **Pictures** and so on – as well as the top level of your **Home** directory. If you want to include other folders – including those on secondary drives – click the Search Settings button and click Add Location next to Custom Locations to add them.

Either search tool requires you to enter keywords for your search – by default, the entire file and its

contents are searched, but you can restrict to search just filenames, plus apply other filters by clicking the settings button inside the Search Everywhere or Search Current Folder box. Here you'll find two drop-down menus: When lets you pick a time from one day to five years ago; What allows you to restrict searches to either files or folders, or to a specific document type.

Tweak Files

You can open multiple *Files* windows at once or press Ctrl+T to open a new tab within the same window. You can then drag and drop files and folders between windows and tabs – the same rules apply here as in Windows, so files are moved by default between folders on the same drive unless you hold the Ctrl key.

Further options can be found when right-clicking files, folders or empty space with the *Files* window. Look out, too, for the hamburger icon at the top of the navigation pane – clicking this reveals further options, such as showing hidden files, altering the icon size and accessing *Files's* Preferences.

Here, you'll find options for displaying files and folders (by default, folders appear before files) and adding context menu items (for creating shortcuts and deleting files permanently rather than sending them to Trash). There's also support for displaying additional file properties when viewing files in grid view.

Customise your desktop

Like all good desktops, Gnome offers plenty of ways in which you can personalise it. The quickest way to access these settings is by right-clicking any blank part of the desktop, where you'll find three relevant options: Change Background, Desktop Icons Settings, and Display Settings. These all whisk you off to different areas of the *Settings* app, which we'll cover in more depth in a future *Linux Basics* tutorial.

For now, we'll focus on the three desktop-related areas. Display is where you go to change the desktop resolution, but it also enables you to switch on a Night Light feature. This allows you to change the colour

» THE LINUX SHELL

There's no avoiding the fact Linux is a command-line based OS. While it's possible to do virtually everything you need from your user-friendly point-and-click desktop, there are times when you have to interact directly with the underlying shell. Don't worry, it's not as scary as it sounds, and next month, we'll take you through the fundamentals of using it.

For now, all you need to know is that you can access the shell using the terminal, a tool that enables you to interact with it using the same language (BASH). This means you can do anything in the terminal you'd normally do from the command line, all without leaving the comfort of your desktop.

You can open the terminal from the Launcher, or save yourself a couple of clicks by using its memorable keyboard shortcut: Ctrl+Alt+T. Once opened, type the commands you need, pressing Enter between each one. For example, the following checks to see if there are any package updates:

```
$ sudo apt update
```

The `sudo` command requires administrative privileges – basically, just enter your user password and hit Enter to

The terminal enables you to reach deep into the heart of your Linux installation.

continue. If you're told that updated packages are available, install them with the following command:

```
$ sudo apt upgrade
```

temperature to reduce eyestrain when using your desktop late at night. You can apply a manual schedule or set it to 'sunset to sunrise', which syncs in with your local calendar. You can then determine the strength of the effect using the Color Temperature slider.

Next is Appearance. Here you can set your desktop style (think Windows desktop theme), a straight up choice between Default and Dark. You can also change the colour scheme from the default orange.

Beneath this are options for changing the desktop background image – there are various themed options based on the numbat, the small Australian marsupial that's lent Ubuntu 24.04 its subtitle (Noble Numbat), but you can select your own via Choose Picture.

The final section – Ubuntu Desktop – provides more practical options for fine-tuning the Gnome desktop. The Desktop Icons section allows you to set a default

size for icons displayed on the desktop – you'll also find a switch allowing you to hide the **Home** folder shortcut.

The next section refers to the 'Dock' – which is actually a reference to the Dash. Like the Windows taskbar and Mac OS dock, you can configure it to automatically hide when any window overlaps it. You can also stop it extending to the top and bottom of the screen, plus change the size of icons displayed on it (useful if you need to pack a lot in). You can even move the Dash to the right or bottom of the screen if you find that more familiar. One final tweak here – Configure Dock Behaviour – enables you to hide mounted drives (including CDs and DVDs) as well as the Trash icon.

Last, but not least, is Enhanced Tiling. Enabled by default, this ensures windows attempt to tile logically around each other. If you don't like this, you can disable all or part of its behaviour. **LXF**

> **QUICK TIP**
>
> Pin frequently accessed folders to *Files's* sidebar for easy access. Simply drag the file from the right-hand pane into the area marked Other Locations and you'll see a New Bookmark option appear.

MANAGE MULTIPLE WORKSPACES
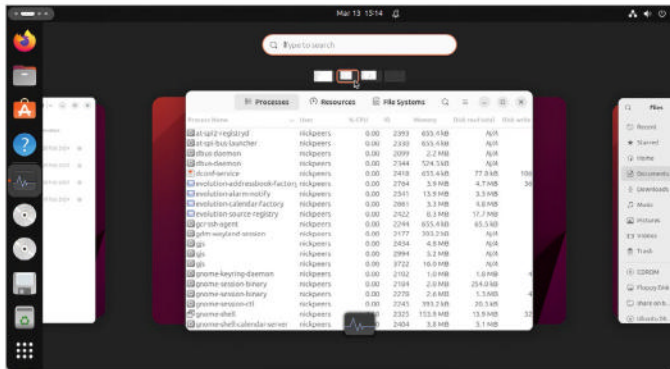
1 View and switch workspace(s)

Click the Activities button in the top-left corner of the desktop to view and manage workspaces. You'll see small thumbnails of each workspace at the top, with your current workspace replicated beneath, its open windows displayed next to each other. Switching is simple: just right-click the thumbnail of your chosen workspace, and you're taken directly to it.
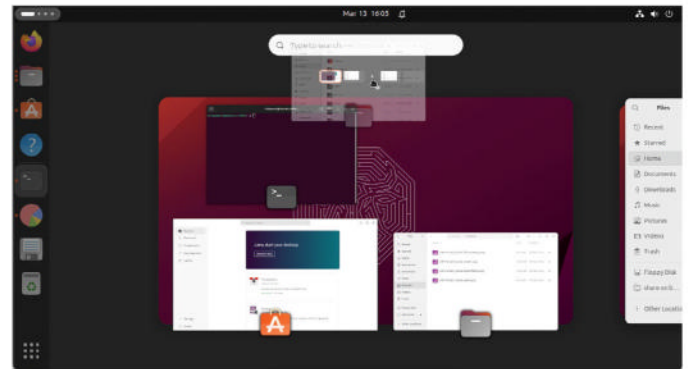
2 Transfer apps between workspaces

If your primary workspace becomes cluttered, it's easy to move any app window to a different workspace. To move it, select its workspace, then click on the window you wish to move and drag it to a different workspace thumbnail to move it to that workspace, or drag it between thumbnails or to the far left or right to create a new workspace to house it.
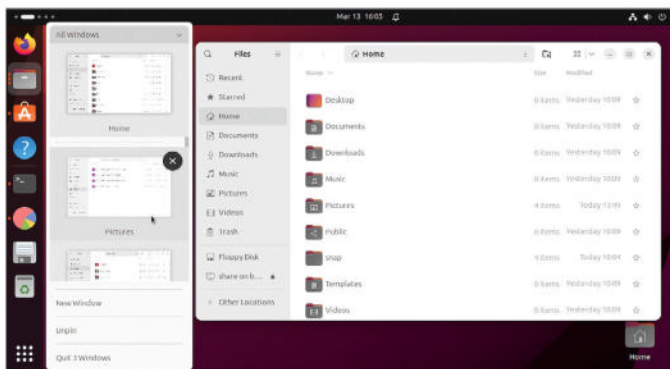
3 Switch apps quickly

What happens when you want to access a window that's in a different workspace? Save time by using the Dash – it displays all open app icons, so clicking one whisks you to that window (and its associated workspace). If an app has two or more windows open across different workspaces, right-click the app icon and expand All Windows to select the one you want.
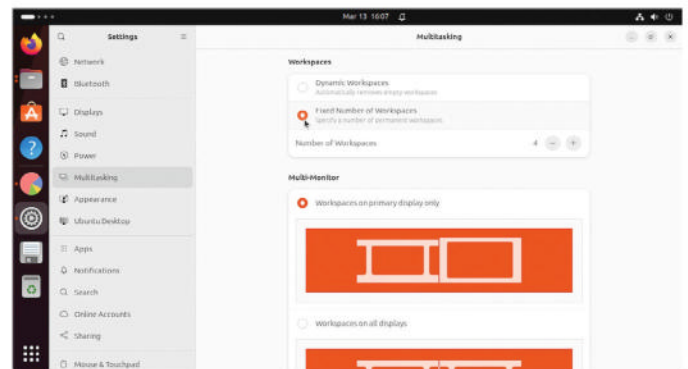
4 Tweak workspace behaviour

Open the *Settings* app and go to the Multitasking section. You'll find options to replace dynamic workspaces with a fixed number of permanent workspaces (you can move windows between workspaces, but not create new ones), spread workspaces across multiple monitors, or change the Dash's app-switching behaviour so it only applies to the current workspace.

» IMPROVE YOUR LINUX SKILLS Subscribe now at **http://bit.ly/LinuxFormat**

BETTERBIRD

Credit: www.betterbird.eu

Enhance email with a better Thunderbird

Avian aficionado **Nick Peers** takes a deep dive into this soft fork of Thunderbird to reveal why you might want to give it a spin.

OUR EXPERT

Nick Peers can still remember the days when email was considered the most advanced form of communication. It may be somewhat passé now, but no social media platform comes close.

Think email client and the obvious example is Mozilla's *Thunderbird.* But whether you're a long-time *Thunderbird* user or looking for a new email client, you may find that *Betterbird* is – if you'll excuse the pun – a better fit. That's particularly true if you're missing features that have quietly been dropped from *Thunderbird* in recent years.

Betterbird is a fork of *Thunderbird,* but it's careful to stick closely to its parent in what *Betterbird* describes is a "soft fork". It closely follows the *Thunderbird* ESR release schedules to ensure it remains as secure as *Thunderbird* itself and is built from the latest version of *Thunderbird* ESR before applying its own fixes, features and customisations.

All of *Betterbird's* bug fixes are submitted upstream to allow *Thunderbird* to benefit from them if it wishes – and many do eventually make their way into the main program. To find out more about the *Betterbird* project in general, check out the box *(below).*

Hatch your better chick

The simplest way to get *Betterbird* on your system is via Flatpak (**https://flathub.org/apps/eu.betterbird.**
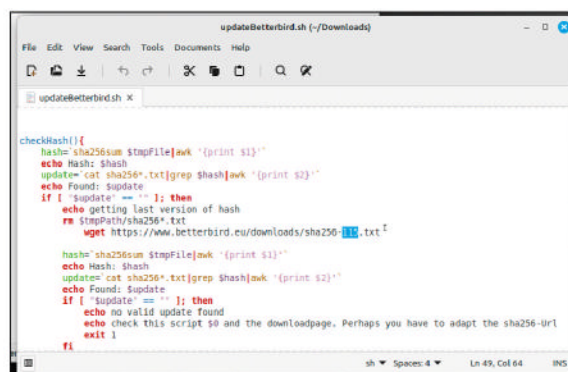
The easiest way to install Betterbird in Debian, Ubuntu or Mint is through the (subtly tweaked) Betterbird Dirty Update script.

Betterbird), although there are glitches to consider, as outlined at its GitHub page (**https://github.com/flathub/eu.betterbird.Betterbird**). ArchLinux users will also find the app is available through its package repo.

Betterbird's native executable is 64-bit only and built on Mint 20.2, which means it's supported on 64-bit versions of Debian (11 or later), Ubuntu (20.04 or

» BETTERBIRD – A BRIEF HISTORY

Betterbird couldn't ask for better credentials. It was created by Jörg Knobloch, former maintainer of *Thunderbird* from 2014 to 2020. Having spent much of that time keeping the project alive as Mozilla's interest focused elsewhere, he was let go under rather unpleasant circumstances – visit **www.betterbird.eu/faq/former.html** for full details.

After being banned from the *Thunderbird* community, Jörg turned his attention to producing *Betterbird,* enabling him to continue development on the email client from an independent

standpoint. It follows the ESR release cycle, so the first patches were released under the version number 91.0b1-bb1 back in July 2021. On 14th July, the first official release – albeit still a beta – saw the light of day with *Betterbird's* first restored feature: the header pane button customise option. It was also accompanied by the first of hundreds of bug fixes to *Thunderbird's* own code, all of which have been submitted upstream for *Thunderbird's* maintainers to incorporate if they wish. And despite the bad blood, *Thunderbird* has accepted

many changes, while others still await approval,

Releases have steadily followed, fixing bugs, changing behaviours and restoring and adding features. Like *Thunderbird, Betterbird* has continued to support older versions even after newer releases have appeared, so the final release of the 91.x series came in September 2022, three months after *Betterbird* 102.0 was released. *Betterbird* 115.0 arrived in July 2023, and at the time of writing, *Betterbird* 115.9 had followed *Thunderbird* 115.9 out of the door on the same day.

later) and Mint (20 or later). You can download and extract this directly from **www.betterbird.eu/downloads** but a better option is to use the *Betterbird Dirty Update* script.

Find out more about the script at **https://github.com/risaer/betterbird-dirty-update** – click the **UpdateBetterbird.sh** link, then the download button next to Raw to save it to your **Downloads** folder. Next, double-click **updateBetterbird.sh** in *Files* to open it in a text editor and make two edits. First, change `lang=de` line to `lang=en-US`. Then scroll to the **checkHash()** section and alter `sha256-102.txt` to `sha256-115.txt`.

Save your changes, close the file, then right-click **updateBetterbird.sh** and choose Properties. Flick the Executable As Program switch to On or tick Allow Executing File As Program under the Permissions tab. After clicking OK, right-click the file again and choose Run As Program (Mint users should double-click it and choose Run In Terminal). All being well, the script will download the latest version, install any updates, then create a shortcut for *Launcher* or your start menu.

First flight

When you first launch *Betterbird,* you're faced with the account setup wizard. It's a simple affair if you're starting from scratch – just enter your email address and password and click Continue to see if *Betterbird* can detect your email settings.

If it can, you're given the opportunity to connect any calendars (including those linked to your email address if you're using a web-based provider like Gmail) and then you're taken to the main screen. If it's unable to detect your settings, you can enter them here – consult your email provider's documentation or support pages for the details you need to enter.

If you're looking to switch from *Thunderbird*, you don't want to faff about with a new profile when you can use your existing one in *Betterbird*. Before going further, visit **www.betterbird.eu/support/index.html#switch-tb-bb** and read the section on switching between the two . Note the guarantee of compatibility only applies to the same version of *Thunderbird*, which must be running on the ESR release schedule (115.x.0 at time of writing) if you plan to move back and forth.

If you're just looking to switch to *Betterbird* and it doesn't automatically pick up your *Thunderbird* profile when you first launch it, it's likely because *Thunderbird* was installed as a Snap (the default in Ubuntu). This means its profile is stored in a different directory from what *Betterbird* is expecting – you'll likely find it inside ~/Snap/thunderbird/common/.thunderbird.

Make sure *Thunderbird* and *Betterbird* are closed, then copy the two folders ending **.default** and **.default-**
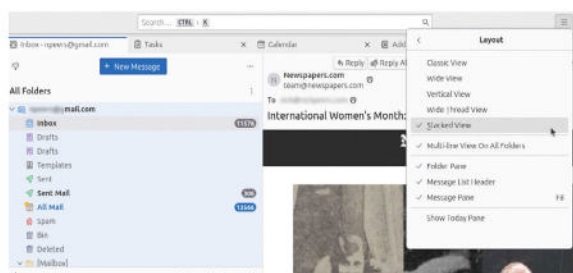
Betterbird offers two exclusive layouts not found in Thunderbird – Stacked is better suited to larger displays.

THE BETTERBIRD INTERFACE
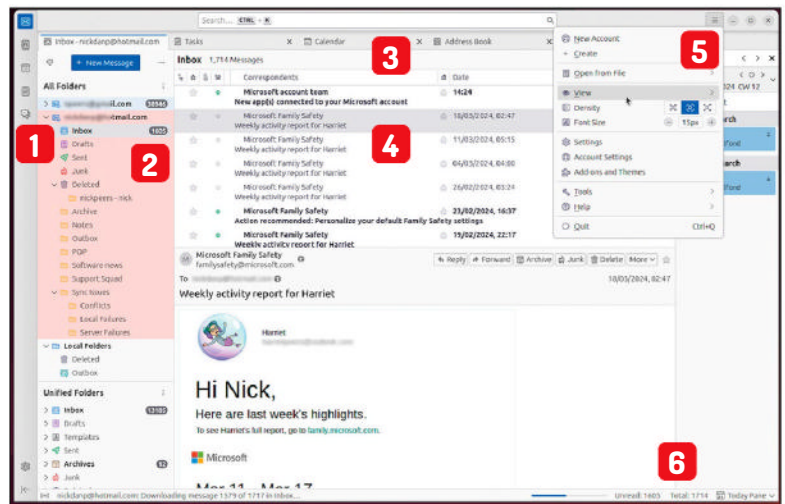
1 **Spaces toolbar**
Provides convenient one-click access to the key parts of *Betterbird,* plus a handy shortcut to Settings.

2 **Folder pane**
Access your mailbox's folders here. A variety of views are available, including Unified (shown here).

3 **Tabs**
Like *Thunderbird, Betterbird* splits its component parts into tabs, making it easy to switch views.

4 **Message pane**
Betterbird offers a range of exclusive views of your inbox – pictured here is its Multi-line view.

5 **Betterbird menu**
Click the hamburger icon to bring up *Betterbird's* menu containing all key options and tools.

6 **Status panel**
This shows the total and unread or selected message count, which was removed from *Thunderbird*.

---

default respectively into your *Betterbird's* profile folder (the hidden **~/.thunderbird** directory). If the folder names are identical, allow *Files* to merge the contents.

Restart *Betterbird* and you should find your old profile is in place, ready to use. If the copied folders had different filenames, open *Betterbird's* profile manager to switch profile. Click the hamburger menu button in the menu bar to open the *Betterbird* menu and choose Help > Troubleshooting Information. This opens a new tab – scroll down to locate the About:profiles link, and click this to open yet another tab. You should see the new profile folders alongside the existing ones. Click Set As Default Profile for the one you wish to use before closing and restarting *Betterbird*.

Language tweets

You should find yourself at *Betterbird's* main screen, which you'll notice looks different from – and more colourful than – *Thunderbird's*. However, before we explore further, you may want to switch from US to British English. There's no native UK English build of *Betterbird*, but a language pack can be found at **www.betterbird.eu/downloads** – right-click the Language Pack link next to English (en-GB) and choose Save Link As to save the pack (an XPI file) to **Downloads**.

Now switch back to *Betterbird*. Open the *Betterbird* menu and choose Add-Ons And Themes. Click the settings cog next to Personalize Your Betterbird and choose Install Add-On From File. Select the XPI language file and follow the prompts to add it.

Once done, open *Betterbird's* menu again, but this time choose Settings. Scroll down to the Language section under General, where you'll see English (US) is

---

QUICK TIP

If you're struggling to get to grips with using regular expressions in your *Betterbird* message searches, bookmark **https://regexr.com** – this website teaches you all about regular expressions, plus explains the basics of building and testing them.

---

selected – click this and choose English (GB) before clicking Apply And Restart to switch to British English.

Ruffling your feathers

As the annotation *(previous page)* reveals, the interface should feel familiar to long-time *Thunderbird* users, and won't take long to master for newcomers. That said, *Betterbird* opens by applying different default views from *Thunderbird*. First, it chooses the traditional Table view for your inbox, with emails displayed in single rows. And second, *Betterbird* places the preview pane displaying the currently selected email beneath the main inbox as opposed to the right of it.

At this point, these are simple layout choices – you can, for example, easily reinstate *Thunderbird's* Cards view by clicking the Message List Display options button next to the Quick Filter button. There's also a handy Sort By menu that allows you to change the sort order (*Betterbird* follows *Thunderbird* in displaying the oldest messages first; select Descending from the Sort By menu to reverse the order).

Further layout options can be found hidden away on the *Betterbird* menu under View. It's here you can switch between *Betterbird's* chosen Classic View and *Thunderbird's* preferred Vertical View for displaying the preview pane. A third view – Wide – widens the preview pane by truncating the folder pane. You can find Wide in *Thunderbird,* too, but the next two options are exclusive to *Betterbird.*

The first is Wide Thread View, which pretty much does what it says on the tin: it's Wide View, except the thread pane is widened to fit the entire window instead of the preview pane. The second is Stacked View, which arranges the thread pane above the message pane on the left, giving over the entire right-hand side to the preview pane.

Birdy view

One of *Betterbird's* headline features concerns its inbox pane. One of *Thunderbird's* longest-standing feature requests has been to support a multi-line inbox view, but it chose to implement Cards view instead. *Betterbird* gives you the best of all worlds – it supports
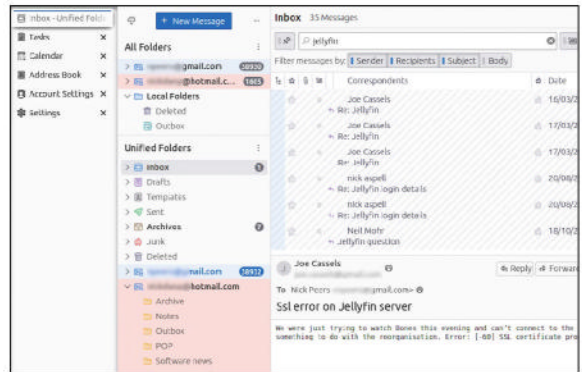
You can choose to tile your tabs vertically in Betterbird – do so from the General tab of Settings under Language & Appearance.

Cards view, but adds its own implementation of multi-line view, which is enabled simply by ticking Multi-Line View On All Folders from the View > Layout menu.

This should place the subject line on its own line directly beneath the Correspondents line – if, for any reason, it doesn't line up perfectly, make sure the blank column header (the Subject header) is between the Read Status and Correspondents headers.

Multi-line support also comes with the ability to rethread messages. This works with single messages only – you can either drag one on to another message (which becomes its parent) or select the message and press Ctrl+T to enter ReThread Mode. Once done, use the arrow keys to place the message where you'd like it to go, and then press Ctrl+T to exit ReThread Mode.

Should you want to unthread a message, you need to select the message from within the thread, then press Ctrl+T followed by Ctrl+X, or press Ctrl+Z to restore the message to its original thread state.

Better plumage

This basically sums up *Betterbird's* core philosophy – rather than reinvent the wheel, it aims to provide you with additional options for fine-tuning the app to your needs. Another example comes with *Betterbird's* support for tabs. When you open your Address Book or Calendar, or visit Settings, they're opened in a new tab, just like your web browser. *Thunderbird* displays these horizontally in the traditional way, but *Betterbird* can be configured to display them vertically in a separate pane that's placed immediately to the right of the Spaces toolbar – this requires a trip to Settings > General, where you'll find Show Tab Bar Vertically is an option under Language & Appearance.

If you monitor multiple email accounts, another advantage of switching to *Betterbird* is its support for colour coding each account to make them easier to identify. This is done on a per-account basis using its Account Settings screen. You'll see a colour bar next to the Account Name – it's black by default, so click to choose a colour from Ubuntu's Choose A Colour dialog. After clicking Select and returning to your mailbox, you'll see the colour is applied as a background to the account's folders in the folders pane, making each one easy to differentiate from the others.

Powerful flight

Betterbird offers the same two mail search tools as *Thunderbird,* but with improvements added. There's

» SYSTEM TRAY SUPPORT

One of *Betterbird's* big selling points is its support for an enhanced system tray icon. This means you can keep the client minimised and receive updates via your menu bar or system tray. The icon works across all major desktops, although Gnome is the worst supported.

The icon should appear permanently in the system tray. You can minimise *Betterbird* to the system tray by ticking When Betterbird Is Minimised, Move It To The Tray under Settings > General. In Gnome, first install and activate the AppIndicator and KStatusNotifierItem Support extension via **https://extensions.gnome.org/extension/615** (at time of writing, we're still waiting for it to add Gnome 46 support to work in Ubuntu 24.04 LTS). Once in place, you need to double-click the system tray icon to bring *Betterbird* back into focus.

The final feature is available to all desktops except Gnome. This provides the system tray icon with a tooltip – roll your mouse over it and it displays a summary of how many unread messages you have, complete with folder list. It's not all bad news for Gnome users, though; the app icon in the Dash alerts you to any unread messages.

support for using regular expressions in searches, plus you can build more complex search terms using groups. Both improvements are outlined in the step-by-step guide *(below),* while *Betterbird* is also capable of searching within encrypted messages.

The quick search tool – always accessible from the menu bar – is usually sufficient for most needs, however. All you need to do is input a few keywords and *Betterbird* displays a list of matching names and email addresses, plus offers to display a list of results – click the Messages Mentioning entry in the drop-down results to view these in their own tab.

Betterbird sorts these results by what it deems to be relevance (click the Sort By Relevance drop-down to change this to by date). You'll see a Toggle Timeline button – click this to use the bar charts to narrow the results to a specific week or day. On the left you'll see more useful filters: From Me, To Me and Attachments at the top; matching people and folders below. You can

leave the tab open for future reference, or close it when you're done.

And there's more
Browse to **www.betterbird.eu/#featuretable** for a complete list of major differences between *Betterbird* and *Thunderbird.* Having carved its own identity, *Betterbird's* developers have changes planned for future releases, including support for calendar subtasks and the ability to compose messages in a tab.

There are glitches – not only does *Betterbird* have to contend with *Thunderbird's* bugs, but it has a few of its own. You can track these at **https://github.com/ Betterbird/thunderbird-patches/issues** where you may find workarounds suggested in the comments. You'll also see a list of wanted features, some of which are in development, while others are on hold due to needing funding. Details for donating to *Betterbird* are on the initial start page and the main website. **LXF**

USE BETTERBIRD'S EXCLUSIVE SEARCH TOOLS
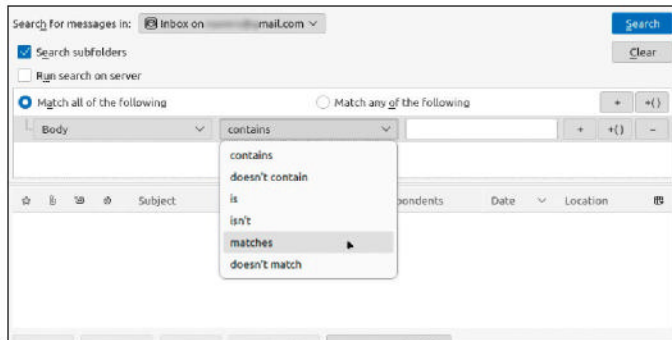
1 Regular expressions
Betterbird's exclusive search tools are accessible via the advanced Search Messages tool (press Ctrl+Shift+F while in the Message tab to open it). First, there's support for regular expression operators to help produce more focused searches. These require you to select either Matches or Doesn't Match as your search type.
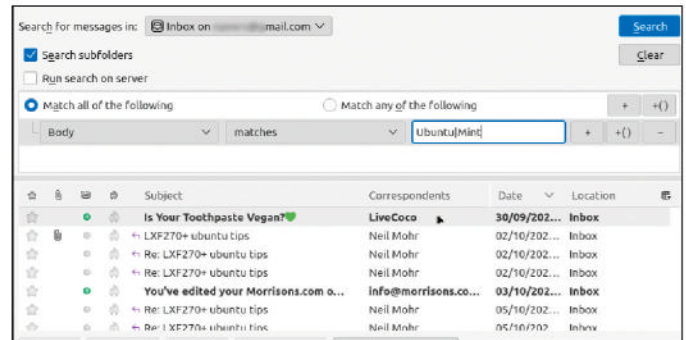
2 Example expressions
The following example finds all emails containing either 'bat' or 'cat' in the body of the email: `^[b-c]at$` – the ^ and **$** strings denote the beginning and end of the match, otherwise matches would include words containing 'bat' and 'cat', such as bath or scat. You can also use regex to search emails containing one or more words: `Coffee|Tea|Water` .
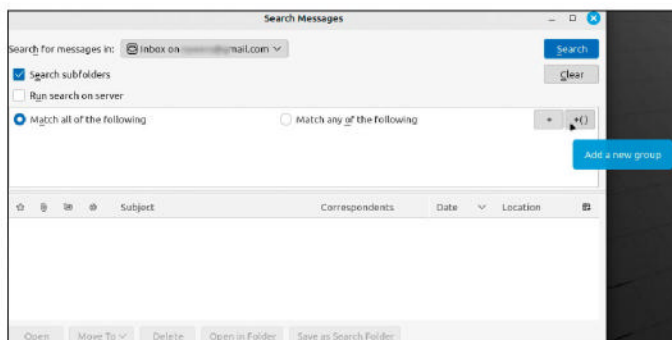
3 Nested searches
Another exclusive feature to *Betterbird* is its ability to combine multiple search terms by nesting them into groups. Each group contains its own conditional matches, which must be met independently of the others. Instead of clicking + to create a new condition, click the +() button next to Match All Of The Following or Match Any Of The Following to create a new group.
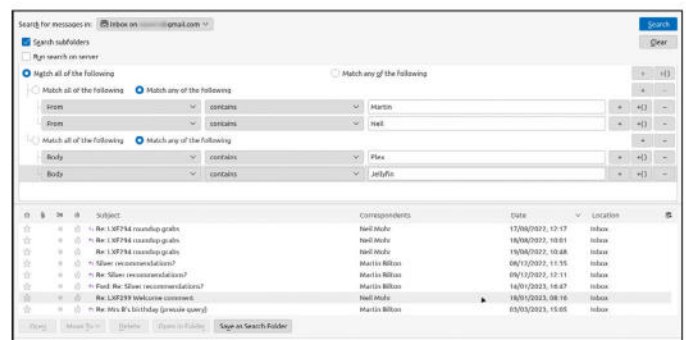
4 Example nested search
Click + within a group to create a new rule within that group, or +() at the top to create a second group. To see how this works in practice, view the screenshot above. This example basically returns results on searches that match two grouped searches, so any emails containing one of Martin and Plex, Martin and Jellyfin, Neil and Plex, or Neil and Jellyfin will be displayed.

KRITA

Using Krita for basic photo manipulation

Amateur photographer extraordinaire **Neil Mohr** finally puts down his GIMP and picks up a Krita for the first time.

OUR EXPERT

Neil Mohr has been playing with analogue and then digital photography all his life.

C an you believe *Linux Format* has never run a *Krita* tutorial, despite many reviews over the years? *Krita* has finally turned v5 and we're taking a flying tour of its features. It's aimed at digital artists, but we're coming to *Krita* from a photography angle, which is more likely how most readers want to use it. This also helps to introduce the *Krita* interface without getting bogged down in its more complex drawing features aimed at artists, and it's likely you'll be more familiar with similar tools in *GIMP* and other photography tools. See the box top-left for installation.

Opening *Krita* for the first time isn't very impressive; you'll want to click the open file and choose a suitable image before the interface springs into life. Take a

gander at the annotation *(below-left)* for highlights of the interface.

Ironically, *Krita's* crown jewel is the digital-artist-targeting pop-up brush palette. We're largely ignoring this, which is a shame as it's a delightful bit of interface design. Hopefully, you'll appreciate just how versatile and functional this would be if you were using a tablet with stylus or a dedicated graphics tablet, enabling you to instantly pop it open, adjust your drawing tool and mode, then effortlessly dismiss it. See the boxout *(on page 63)* for more on brush control.

When it comes to adjusting photographs, there's a few jobs we find ourselves doing over and over. The first is fiddling with colours, largely contrast, saturation and perhaps the colour temperature on indoor shots. We also often clone small areas to touch up images, alongside area selection, so we can clone or target adjustments to specific parts of a photo. We'll look at how these areas work in *Krita* before walking you through adjusting a landscape photograph.

THE KRITA WORKSPACE

1 Toolbox
Located on the left-hand side, the toolbox contains various tools for painting, selection, transformation and more.

2 Colour selector
At the top-right, you can find the colour selector, enabling you to choose the colours for your brushes and other tools.

3 Layers
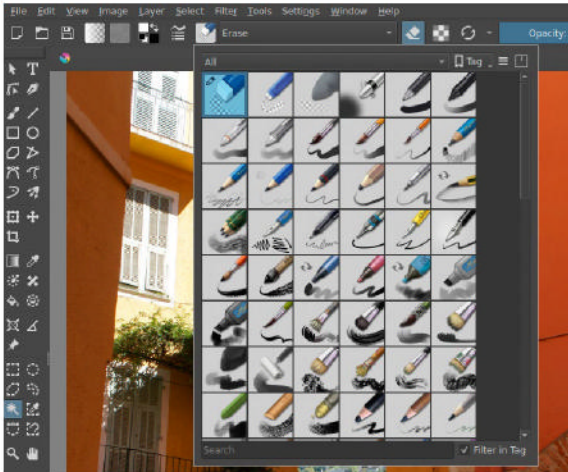Located on the right-hand side, the layers docker displays all the layers in your

document, enabling you to organise and manipulate them.

4 Brush presets
Directly below the colour selector, you'll find the brush presets docker, which is where you can select and customise brushes for painting and editing.

5 Pop-up palette
This powerful UI element is amazing for tablet and stylus users, but it's just as handy for those with a mouse to quickly access settings.

Not too bright!

Rather than a discussion of the intelligence of **LXF**'s editor, this section is about *Krita's* colour controls. These are the same array of controls as you find in *GIMP* and most image editors, such as Brightness/Contrast, Levels, Curves and Saturation.

Interestingly, *Krita* doesn't have a straight Brightness/Contrast control, but it does have an Auto Contrast adjustment that you can find – along with the other colour controls – under the Filter > Adjust menu.

If you want to balance the black/white contrast levels, we suggest using Levels. Levels let you more easily adjust the overall tonal range of a photograph, balancing the black and white points. It's such a fundamental element of photo editing that it should be one of the first things you apply to a new image. Press Ctrl+L to bring up the classic Levels controls, then drag the left-hand Black and the right-hand White levels to enhance the contrast.

Do note the grey and colour mix icons top-left; click the colour one to see the mix of RGB over the intensity range. This also activates an individual channel selector, including Hue, Saturation and Lightness controls. To the right is a double-arrow icon

There's a huge selection of pre-defined brushes available in Krita, many of which ape natural materials.

that resets any changes and below this is a magic wand icon, which activates the Auto Levels mode.

Finally, we'll mention the Create Filter Mask button at the bottom of the dialog. This is a general feature and clicking it creates a new effects layers using whichever filter is open when you click it. It switches you to drawing mode, where you can apply an alpha mask in the active layer – use the Layer Docker on the right to select the original image layer.

If you want to boost the brightness, try the curve tool called Colour Adjustment Curves. The name curves comes from the ability to adjust the intensity of colours across the brightness range on a customised curve through the use of control points. This makes it easy to increase the brightness of dark areas, maintain mid-tones and reduce the levels of the brightest areas in a natural way, for example. Shifting this to specific colour channels, you're equally able to do this with the individual colour ranges, so boosting the red, green, blue or a combination of these within an image. Press Ctrl+M to open the filter; the easiest approach is to left-click and drag the middle of the curve up. Click on the curve to add another control point and press the Delete key to remove the currently selected one.

A final colour control to note is the HSV Adjustment filter – press Ctrl+U. This is similar to *GIMP's* Hue-Chroma control, as at a basic level it provides hue shifting, saturation and lightness control, which connects to the HSV (Hue, Saturation, Value) name, with the other TLAs being other colour model modes. It provides a quick way to tweak hue and saturation.

Clone heads

We're not going into detail here on *Krita's* amazing brush engine, but if we look at its Clone tool, that'll do half the job for us as it's part of the brush system. Cloning is a regular trick used to clean up blemishes and cover up unwanted objects in photographs. Most art packages have a shortcut in their toolbar but not *Krita.* Select the Freehand Brush tool – third icon down on the left in the toolbar – or press B. Use the bottom-right Brush Presets Docker, click Search, type `Clone` , then select it. Hold Ctrl and left-click the area you'd like to clone, and you're good to go.

Right-click and click the right arrow icon to access the quick pop-up controls: flow, brush size, opacity and

» A CROSS-PLATFORM ANIMAL

Getting *Krita* is amazingly easy. Its cross-platform support is a combination of delicious open source code and successful commercialisation. There are, of course, Windows and Mac OS downloads alongside Linux, but also worth noting are Android, Chrome OS and Haiku releases. When it comes to Linux, Ubuntu offers the latest *Krita* release (that's 5.2.2 at this time) via Ubuntu Software as a Snap. If you want the Flatpak, it's in Flathub (**https://flathub.org/apps/org.kde.krita**) and you can install it with:

```
$ flatpak install flathub org.kde.krita
$ flatpak run org.kde.krita
```

You can also grab a 64-bit AppImage from **https://krita.org/en/download**. Assuming the file is in **Downloads**, make it executable:

```
$ cd ~/Downloads
$ chmod a+x krita-X.X-x86_64.appimage
$ ./krita-X.X-x86_64.appimage
```

If you want to support *Krita* on the simplest level, you can buy a copy via various digital download stores, such as Steam, Epic, Microsoft and the Mac App stores. But to maximise the money that goes to the developers, you should donate direct at **krita.org**.

fixing the clone point. Full brush settings are accessed via the top icon bar, Setting being the seventh icon along, next to the foreground/background colour icon. This gives you crazy control over the shape, force and effect of the brush. Try turning on Healing mode – this automatically blends the colour and edges of the cloned area for a more subtle result. See the boxout on *Krita's* brushes *(page 63)* for more details on these settings, though they're less useful for cloning.

Super selections

The final area we're looking at is making selections. It's vital to be able to select areas within images so you can isolate them for editing, cutting out or masking, to protect them from changes you're making to the rest of the image. As you would hope, *Krita* offers a range of selection systems, going from basic, quick and easy through to more advanced vector-based brush paths that can be created, modified and saved.

To start off easy, the basic selection tools can be found at the bottom end of the left-hand toolbar; all
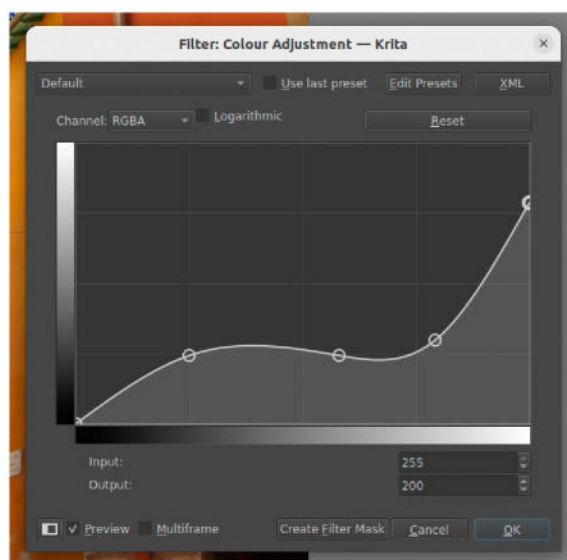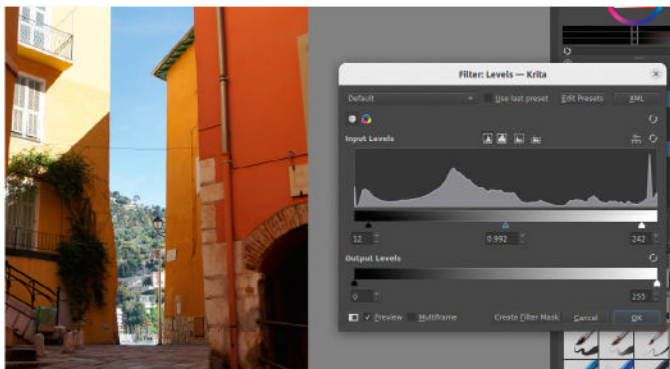
The classic Curves is available to adjust image brightness.

LAVISHING YOUR LANDSCAPES

1 Tweak contrast

In this step-by-step guide, we're going to have a little play with enhancing a landscape image. To begin, open your image in *Krita* and switch to the Levels with Ctrl+L. Ideally, you want the left Black and far-right White controls moved to where the intensities first start to rise this will best balance out the contrast range.
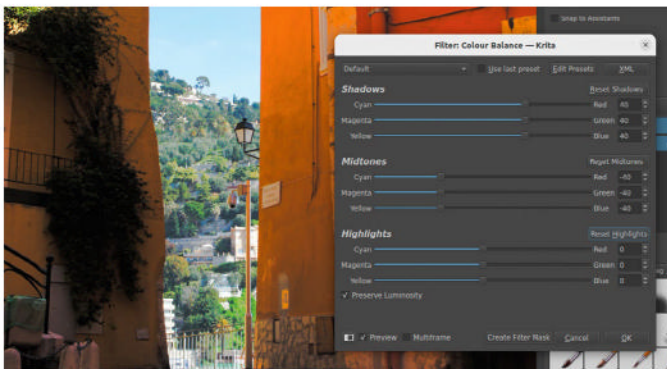

2 Colour balance

To tweak the colour balance, press Ctrl+B (or go to Filter > Adjust > Colour Balance) and use the sliders for Shadows, Midtones and Highlights to enhance the overall colour of the photo. Increase the saturation of warm tones to make the sunshine more vibrant, or adjust the blue tones to create a cooler atmosphere.
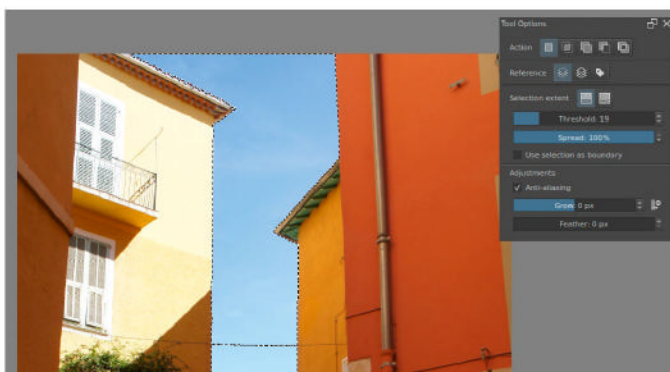
3 The sky is falling

Use the Magic Wand tool to select out the blue sky. Click once in the middle – it probably only selects a band, so hold Shift and click further up and repeat. Press Ctrl+C and then Ctrl+P. A new layer is created with just your selected area within it. This enables us to process the sky on its own without affecting the original image at all.
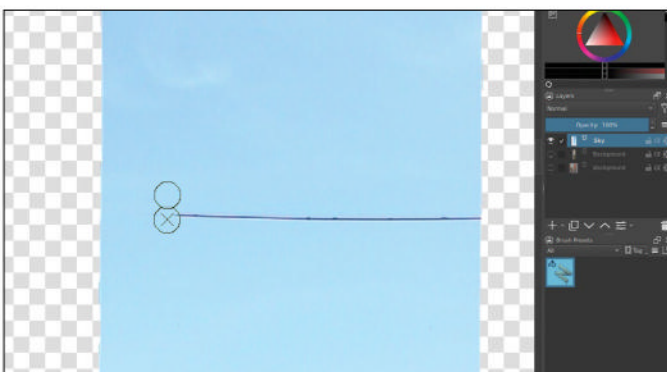
4 Clone away

Let's remove that wire because it's an eyesore. Use the search bar in the Brush Preset Docker (bottom-right) and type `Clone`. Make sure it's selected, alongside the standard Brush tool in the toolbar. Zoom in and hold the Ctrl key and click on a blue area. This first circle is the clone area; the second crossed circle is your drawing brush – adjust the size to suit.
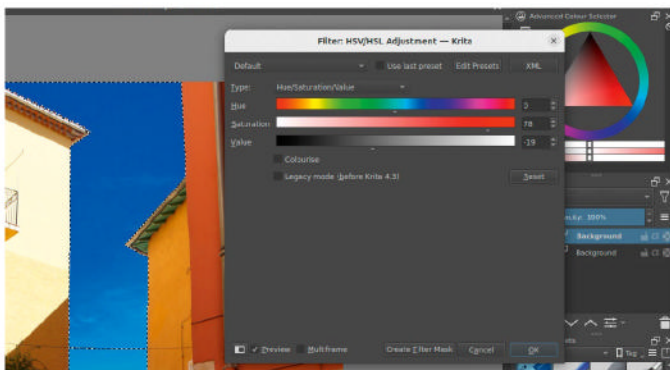
5 Hue and saturation

Press Ctrl+U (or select Filters > Adjust > HSV Adjustment), switch the Type to HSV (Hue, Saturation, Value) and ramp up the Saturation, while reducing the Value a touch. This produces a much richer blue sky. Switch the Type (the colour model) to see how that affects the output because there's a significant difference between colour models.
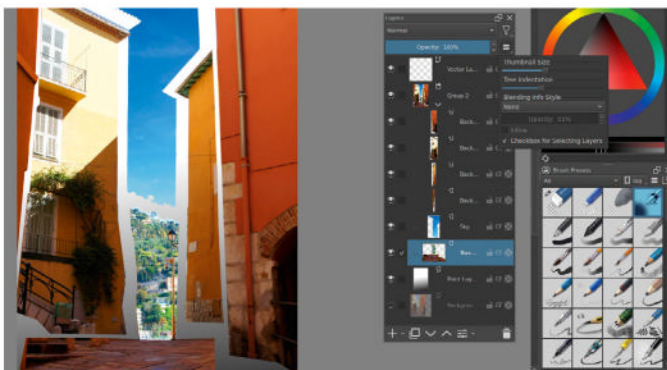
6 More selections

Take time to experiment with the selection tools and build your muscle memory for using the navigation controls. Cut out more elements and play with the Layer Docker. Increase the thumbnail size and create a group (this enables you to hiding/viewing all layers). Remember to save the project as a *Krita* document rather than PNG or JPG.
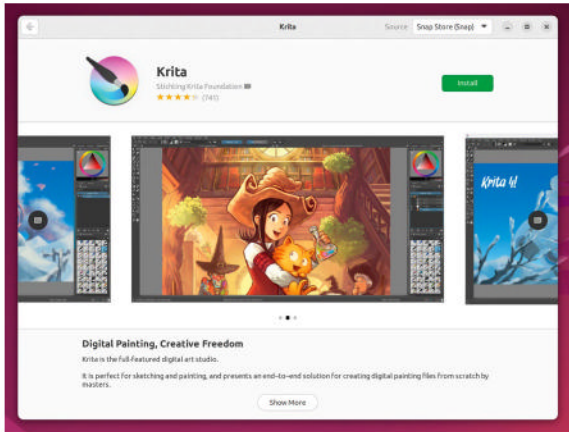
You can find Krita well supported in almost all software stores.

>> MASTER YOUR BRUSH

Brush control is essential for unleashing your creativity and achieving precise results, so let's delve into the brush controls and how to use them. *Krita* offers a wide range of brush presets tailored for different artistic styles and tasks. You can access these from the Brush Presets Docker, bottom-right. Jump in and experiment – there's a full range, from pens, pencils and erasers to clone brushes.

With a brush selected, press F5 to access the Brush Editor. Here, you can adjust a phenomenal number of parameters, such as size, opacity, flow, spacing and shape dynamics to create a brush that behaves exactly as you'd like.

There's no way we can cover all the brush engine settings here – there are 18 different brush engines, such as chalk, bristle, clone and so on, all with their own complex settings. There's documentation on all the settings at **https://docs.krita.org/en/reference_manual/brushes.html** – but we will say that diameter is the size of the brush, and don't overlook the right-hand scratchpad to test your changes.

To streamline your workflow, *Krita* enables you to manage and organise your brush presets efficiently. You can create custom brush presets, group them using tags, and import/export presets for easy sharing or backup. Use the default My Favourite tag to list brushes in the pop-up palette, but it only shows the first 10 by default.

their icons have dotted lines, the first four being rectangle, oval, polygon and freehand selections. Once made, if you hover over the selection edge, you can left-click and move it. It's possible to add to and remove from a selection but it's pretty rough and ready.

Adding a little flexibility to the selection process are the bottom two: the Bezier and Magnetic selection tools. In between these two groups are the classic Magic Wand and Similar Colour selection tools. Both effectively use a threshold level for selecting areas of similar colour. Use the Tool Options Docker on the right to adjust this and the overall area you're selecting.

At first, it appears you can't save or easily edit a selection once it's made – that is, until you discover Selection Layers. Roughly make a selection, then in the Layer Docker on the right, click the drop-down menu by the lower-left + icon and choose Add Local Selection. A new alpha channel appears containing the area you've selected. Select the Background and hide the Selection Layer – the selection is cleared. Make it visible and the selection reappears. Select the Selection Layer and you can directly draw on it, modifying the selection. Select the Background, hide the Selection Layer and you can make a new fresh selection and continue editing as you need be. Just create more Selection Layers as you need them.

The Selection Layer does provide a powerful editable selection system, but it's still not quite like paths. *Krita* doesn't offer an equivalent *Photoshop*-like path selection tool, but what it does have is a powerful Vector Layer system on which you can use the Freehand or Bezier Path tools. These are primarily aimed at creating scalable shapes and art, but you can convert them to selections and so use them to isolate shapes on the background image and then manipulate and edit the path further at any point.

If you want to create a selection this way, click the drop-down menu by the lower-left + icon in the Layer Docker and choose Add Vector Layer. Select the Freehand Path tool from the standard left-hand toolbar; it should be the sixth icon down. Make sure you have a suitable brush selected – a couple of pixels wide at most – and draw the selection you would like.

With the line drawn, click the Edit Shapes tool – left toolbar, second row down – then click on the vector

line you've drawn and edit and move it to your heart's content. Once happy, click the top-left Select Shapes tool icon. Click inside your shape, then right-click and select Convert Shapes To Vector Selection. Click the drop-down menu by the lower-left + icon and pick Add Local Selection. Boom! You have a normal selection and a path to go with it. Click the Background Layer, select Copy, and your selection is copied out of the image. It's also possible to convert the Vector Layer to a Selection Layer by right-clicking on the Layer Docker.

By this point, you should have the basics down for colour correction, patching up and cloning areas, alongside making good selections so you can protect and isolate areas for editing. We think you'll find *Krita* is easy to pick up and offers a truly polished, professional interface with some sensational tools for both photography and digital artistry. **LXF**

QUICK TIP

The official *Krita* forums are the ideal place to get help, advice and inspiration from other artists and the wider *Krita* community. Head to: http://krita-artists.org
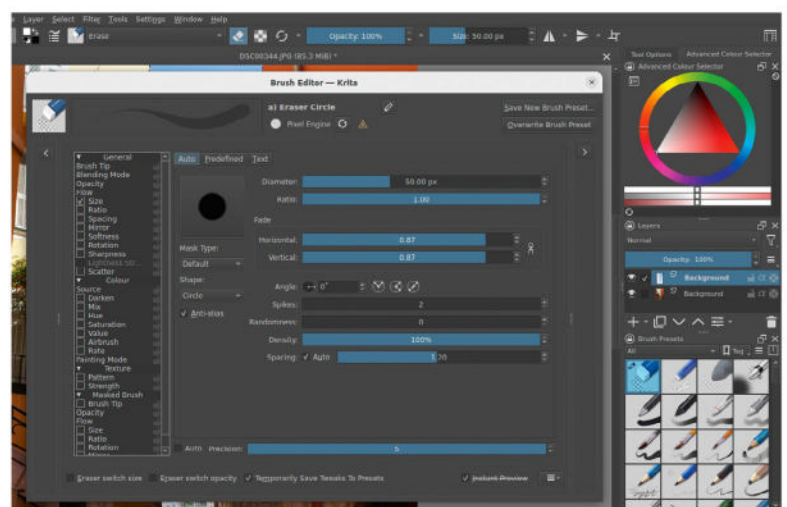
The brush engine options can be rather overwhelming.

>> SEE MORE OF NEIL'S HOLIDAY PHOTOS Subscribe at http://bit.ly/LinuxFormat

BACK ISSUES » MISSED ONE?

ISSUE 314
May 2024

**Product code:**
LXFDB0314

In the magazine
Join us as we explore Linux's infiltration of Windows, and discover how you can also get Windows apps running in Linux as we uncork the latest *Wine* release. And let's not forget a *Roundup* of time trackers, tutorials on vulnerability audits, setting up a home ebook server, streaming games, and more, along with distro reviews, news, and packed Pi and *Adminsteria* sections.

ISSUE 313
April 2024

Product code:
LXFDB0313
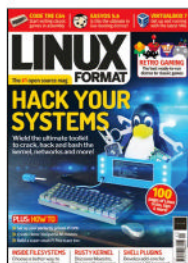
In the magazine
Discover how to use the ultimate hacker's toolkit, staying out of trouble while doing so. And join us as we take the Puppy Linux developer's new distro for a run and explore its container features. Plus, we have a *Roundup* of retro-gaming distros, an in-depth look at filesystems, tutorials on adding plug-ins to our **LXF** shell and getting more from your VMs, plus news, reviews and oodles more.

ISSUE 312
March 2024

Product code:
LXFDB0312

In the magazine
Blast off into the future with a look at the five best next-gen distros, and discover whether Raspberry Pi or Orange Pi is the best SBC for you. Plus, learn how to rescue retro media, add NPCs to your own point-and-click adventure, emulate an analogue computer, and lots more. We've also squeezed in hardware and distro reviews, a password manager *Roundup*, news and more.

ISSUE 311
February 2024

Product code:
LXFDB0311
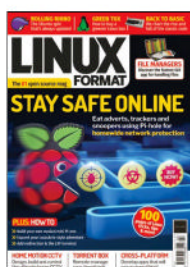
In the magazine
Stay safe online by sending net nasties to *Pi-hole*, and read about the rise and fall (and rise again) of Basic, and the roll-out of the rolling-release Rhino Linux distro. You can also learn how to strengthen your shell history, set up a home CCTV system, add an inventory to your point-and-click adventure, and much more. Plus, check out our packed reviews, news and Raspberry Pi sections.

ISSUE 310
January 2024

Product code:
LXFDB0310

In the magazine
Learn Linux with our in-depth guide to getting started with Elementary OS, or explore a more security-centric distro as we dive into Tails. Plus, find out how to perfect the art of OCR, finish your *WordPress* website, recreate the BBC Micro, and make your own point-and-click adventure. And no issue of *Linux Format* would be complete without news, reviews, Pi projects and much more besides.

ISSUE 309
December 2023

Product code:
LXFDB0309

In the magazine
Manage your apps by harnessing the power of *Podman,* and discover how random numbers are used by the kernel to keep your data safe. Plus, find out how to master printing in Linux, customise your *WordPress* website, tweak your kernel scheduler, use Python to scrape the web, process your smart-home data, and much more. And make sure you read our *Roundup* of Raspberry Pi killers…

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.

C64TAPEDECODE

Credit: https://github.com/lunderhage/

TAP up your old C64 cassettes

Christian Cawley archives some old Commodore 64 cassettes using a handy USB adaptor and popular open source audio software.

OUR EXPERT

Christian Cawley's retro collection includes an original Commodore 64 and a VIC-20, along with an old Ingersoll TV sports system. None of them have been switched on since the '90s.

Old data should be backed up, and it's becoming easier to read files created and saved on other platforms. Thanks to emulation and advances in how data is read on old formats, you can back up cartridges, cassettes and even disks, from 8-bit and 16-bit systems. Here we're looking at C64 cassettes.

But why?

We have a ton of old computer cassettes from 1984 to 1991. On there are seven years of data from games, software and over 84 months of accumulated activity.

In some cases, the cassettes are personal fast-load backups made with the help of an Action Replay cartridge. But what we're particularly interested in are the cassettes that hold data from art packages, tools like *Graphic Adventure Creator,* and other project work.

The reckoning was that if our old Commodore 64 cassette player/recorder (known as a C2N Datasette) still worked, it should be able to copy the data from the cassettes, create images of the data on our Linux system, and review what we were up to 40 years ago. After all, those cassettes might be doing well, but they're not going to last for ever.

Most specifically, we're interested in the contents of the tape holding *The Mystery of Silver Mountain,* a game published in type-in form by Usbourne (and still available in PDF form: **https://bit.ly/LXF315-usbourne-books**). Our recollection of this endeavour is that some errors found their way into the code, and the desire is still there to fix this so that it can finally be finished.

The process, then, is to find an adaptor, a working Datasette, and some software to convert the data (stored as audio on a cassette) to a TAP file. This should then be compatible with the *VICE* Commodore 64 emulator, or a C64 Mini – if not, inverting the audio should do the trick.

The cable adaptor

The first task is to get hold of an adaptor that can convert the signal from the C2N Datasette to USB.

Various approaches have been taken to achieve this, but our favourite is the build based on the TrueTape64 (**https://bit.ly/LXF315-truetape64**). This relies on an Atmel AVR ATTiny2313 microcontroller, and the schematics and parts can be found on the
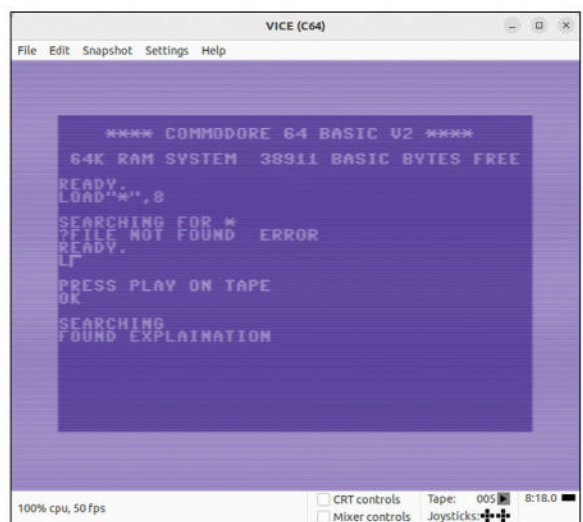
The VICE emulator is capable of emulating a Commodore PET, VIC-20, Commodore 64, Commodore 128, and even a Plus/4.

project's GitHub. It is possible to find assembled versions online (we got ours on eBay, **https://bit.ly/LXF315-ebay**) or via the designer's website (**https://bit.ly/LXF315-load64**), although import duties may impact availability.

One other thing to note: if you own a Commodore Datasette, the cable adaptor outlined above is the tool to use. If, however, you own a cassette player with an audio-out cable that you can connect to your PC, this can also be used. Better still, you don't need to worry about finding an increasingly rare cable adaptor on eBay or Etsy.

Clean up the cassette player

Before you even think about creating an image file from a cassette, give your cassette player a thorough check. If it has been on a shelf or in a drawer or sitting in a box for 30-plus years, it is likely to have attracted dust and even dirt.

In the case of our own Datasette, the device's cassette cover is long since lost, which meant interior cleaning was made easier. Use a can of compressed air, or a powerful vacuum cleaner to suck up anything

loose. You can also employ a ball of adhesive putty or double-sided tape to help clean it up.

Ensure dirt stays away from the tape heads, however; careful attention is required around these. First, press the Play button to push the heads into view. Then use a cotton bud with a dab of isopropyl (rubbing alcohol) and gently clean the tape head.

Keep doing this – changing cotton buds if necessary – until the cotton stays clean. You should also clean the exterior of the cassette player while you're at it, just in case some other dirt ends up inside the device.

Confirm azimuth alignment

It is probably worth noting that getting cassettes to load was never an easy task. The usual problem was alignment issues with the read tape head, which could be adjusted using a screwdriver. There is a small hole on the Datasette, above the second 'o' of Commodore, where a 2mm cross-head screwdriver can be inserted to adjust the alignment.

Of course, this is pretty tricky to do and involves a lot of trial and error. Back in the day, you could use a special Azimuth Alignment cassette that would display a particular colour on an attached TV or monitor if the alignment was correct. Obviously, this isn't an option if you don't have a working C64.

WAV conversion

Once the WAV is created, you need to convert it to a format that is readable by the Commodore 64 emulator (or even original hardware). Several projects have been developed over the years that do this, but from what we can see, none of them has been maintained in the past couple of years (please do correct us on this).

After some research and testing, we settled on *C64tapedecode,* by lunderhage. This is a collection of utilities that cover all bases – converting WAV to TAP, TAP to WAV, creating a TAP file from text, and extracting files from a TAP file. (The one we are using is the WAV-to-TAP converter, *wav2tap.)* Installation requires use of *Make,* (you may need to `apt install build-essentials`) but the software is lightweight, so compiling and installing takes seconds. Start by opening the terminal and cloning the repository:

```
$ git clone https://github.com/lunderhage/
c64tapedecode.git
```

Then, `cd` into the **src/** directory:

```
$ cd /c64tapedecode/src
```

Here, run *Make*:

```
$ make
```

When this completes successfully, run *Make install* as root:

```
$ make install
```

The *C64tapedecode* utilities are now ready to use.

Have the Audacity

As noted above, if you're using a non-Commodore cassette player to record your cassettes to your Linux PC, you don't need a special USB cable. But whether you're using a modern cassette player, a WHSmith cassette player intended for a BBC Micro or ZX Spectrum, or you're using the Commodore C2N 1530 Datasette, you need a tool for recording the audio.

This is where *Audacity* comes in. If you don't already know about the most popular open source audio editing software, you're in for a treat. We've been using it for podcasting and recording audio drama for years, but it has so many other uses; it can record from your soundcard, enabling you to make an MP3 or FLAC copy of every sound playing on your PC, for example. It is also perfect for recording the noisy audio sent from a computer cassette.

Start off by installing *Audacity*. We recommend installing using *Apt* or *DNF* rather than the AppImage or the Snap, as we've personally had some problems with both alternatives on multiple PCs:

```
$ sudo apt install audacity
```

or

```
$ sudo dnf install audacity
```

Once you have installed the software and have it running, connect your player and insert a cassette. In

QUICK TIP

Unwanted Datasettes can be found relatively cheaply on auction sites. Note that the adaptor we used was for the C2N model (for VIC–20 and C64 computers), rather than the 1531 Datasette (for Commodore 16 and Plus/4 computers). Specific adaptors can be purchased for that model.

» WHAT'S ON YOUR OLD CASSETTES?

The situation with backing up cassettes into data files (ROMs) is the same as it is for disks and cartridges. As explained in **LXF313**, if it's your data, created by you, or data you have been given the right to store, you're perfectly fine creating a virtual tape image of the data. Such data can vary from person to person.

Things get muddy when it comes to buying lots of old cassettes. A check on eBay reveals collections of not just used

games for sale, but also used 'blank' cassettes as sold by WHSmith, Boots and others. Of course, these aren't blank, and may hold pirated games or personal data of the original owner, which means that legally you shouldn't make a copy.

In our collection, there's everything from save games and data from spreadsheets to type-ins from old computer books. Then there are graphics, attempts at music, and some

'stories' (but we won't bore you/ embarrass ourselves sharing those).

The point of all this is to encourage you to consider what you're planning to archive, and keep a record of it. If the cassette is labelled, you're halfway there. But it pays to note down the contents of each TAP file you make, perhaps using a spreadsheet to ensure it's all recorded accurately. And if you back up data that isn't yours, don't keep it.

CONVERT CASSETTE DATA TO TAP
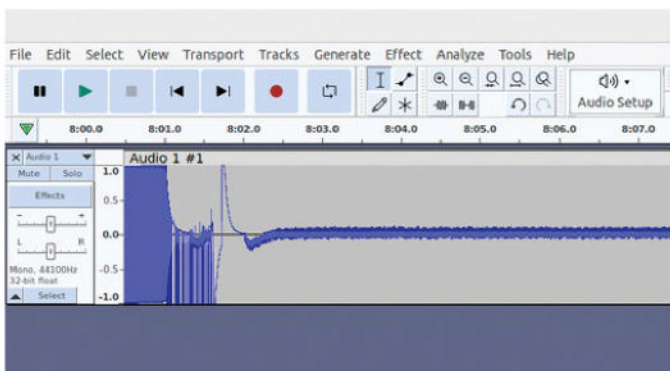
1 Record the audio in Audacity
Start by connecting your cassette player to your computer with a compatible cable. Launch *Audacity* and then use the Audio Setup menu to select the appropriate recording device. With the correct selection made (some trial and error may be required to get this right), hit Record in *Audacity*, then press Play on the cassette deck.
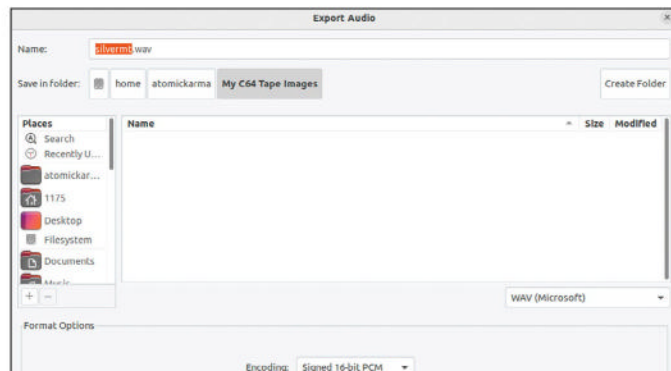
2 Save as a WAV
Wait until the cassette player stops and then press Stop in *Audacity* to end the recording. The default save option in *Audacity* creates a compressed audio project, so rather than using this, you should go to File > Export > Export As WAV. Make sure you give the file a meaningful name, so you will be able to recognise it later.
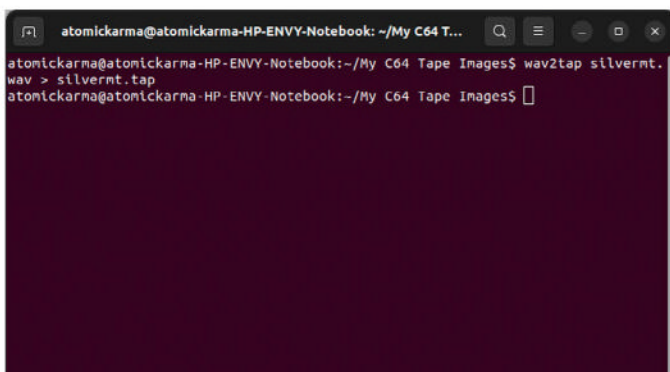
3 Convert WAV to TAP
With your conversion software, input the command to convert the WAV file to a TAP file. In *C64tapedecode*, use the command `wav2tap filename.wav > filename.tap`, taking care to change `filename` to whatever names you are using. Note that you might need to add the **-r** option to invert the audio for emulator compatibility.
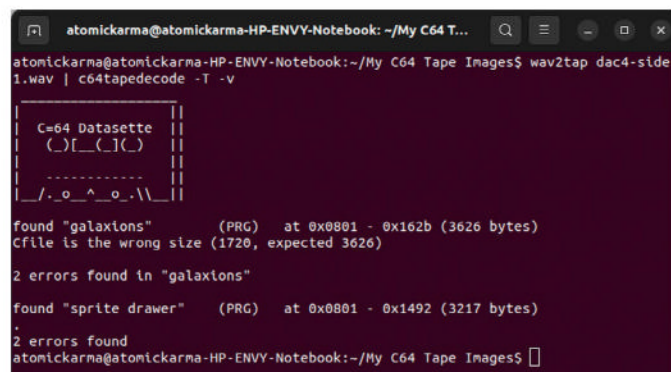
4 Create PRG files
When using *C64tapedecode*, you can also create PRG files. Use the following command:
`wav2tap filename.wav | c64tapedecode -T`
This extracts the files from a temporary TAP file, which can then be opened in *VICE*. You can also add the **-v** option here to see the verbose output.
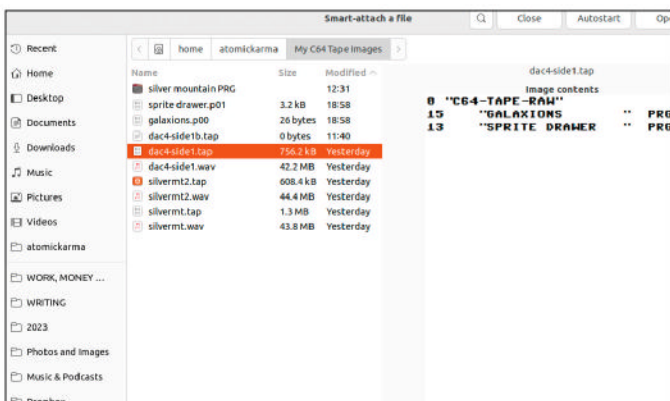
5 Open tape image in emulator
With the TAP or PRG created, it is ready to be opened in the emulation software. Go to File > Smart-Attach Disk/Tape and browse to the tape image file. Select it and click OK. With this option, the virtual tape auto-loads – there is no need to enter a command.
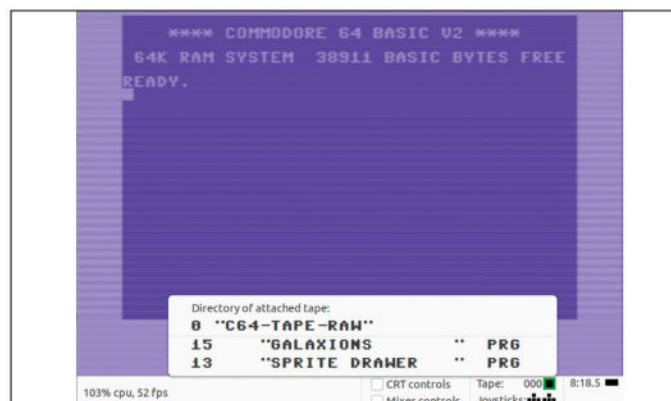
6 Load from the menu
For tapes with multiple programs, you can use the quick menu. At the bottom of the *VICE* window, right-click Tape to select and load programs on the tape image. Note that if software is already running, you should first soft-reset the emulator (File > Reset > Soft Reset).

Audacity, open Audio Setup > Recording Device and ensure the correct audio input is selected. For a USB adaptor, the USB audio device should be selected.

You're ready to create a recording of a magnetic tape storing data from the last century. Press Record in *Audacity,* then press Play on the cassette deck. If everything is working correctly, you should see the waveform as the data plays through *Audacity.*

Leave the cassette playing until the end. When it stops, halt the recording and save the project. At this stage, you should also go to File > Export > Export As WAV to prepare the data for conversion, but give it a meaningful name in case you need to return to it later.

From WAV to TAP
With an exported WAV file ready to convert, you're ready to use the *C64tapedecode* utilities. Open a terminal in the directory you have saved the WAV file, then convert it to TAP format with:

```
$ wav2tap filename.wav > filename.tap
```

This creates the TAP file, ready to be opened in *VICE.* Various additional commands are available, which you can check with the **-h** condition. Of the most relevance at this stage is **-r**, which inverts the WAV. Some tools, such as the 1530USB, automatically invert the audio, so it can be read by the original hardware or an emulator. If this doesn't happen, WAV-to-TAP conversion tools can invert it for you.

```
$ wav2tap filename.wav > filename.tap -r
```

Other tools within the *C64tapedecode* utilities feature their own additional commands. For example, *C64tapedecode* itself can be used to extract files from a TAP file into CBM ROM loader format as follows:

```
$ wav2tap tape.wav | c64tapedecode -T -v
```

Two things to note: first, the **-v** adds verbosity, so details from the process are printed in the terminal. Second, this doesn't create a usable TAP file; this is discarded as the files are extracted. But you can load the PRG file (usually sequentially numbers **.p00, .p01** and so on) into *VICE.*

As things turned out, the cassette we had selected (labelled 'Silver Mountain') turned out to include other data that wouldn't run. It is possible a fast-load system was employed when the cassette was authored, which is something we'll have to come back to later.

However, we have plenty of cassettes, so we selected another. This one was labelled 'DAC/4', the author's father's initials and the fourth tape he compiled. The listing is slightly garbled, but we recognised the game titles from a type-in book (*More Games for Your Commodore 64* by Robert Young) so decided this would be the tape to use. The mysteries of a fantasy mountain adventure would have to wait.

Loading TAPs in VICE
VICE is the *Versatile Commodore Emulator* and is available on all manner of platforms, from Linux and Mac to MS-DOS and even Amiga OS. It emulates C64, C128, VIC-20, most PET systems, and the Plus/4, as well as a few others.

If you don't already have *VICE* installed on Linux, it can be downloaded manually (**https://bit.ly/LXF315-**

vice64) or installed with *Apt, DNF* and *Pacman,* for example:

```
$ sudo apt install vice
```

(It is also available as a Flatpak.)

To load your TAP file in *VICE,* launch the software then open File > Smart-Attach Disk/Tape (or press Alt+A) and browse to the TAP file to select it. This should automatically load the virtual tape in *VICE.*

If you want a more natural Commodore 64 experience, you can opt instead for File > Attach Tape Image (or Alt+T), which is the digital equivalent of inserting the cassette into the Datasette. At this point, input LOAD or L then Shift+O and hit Enter. At the PRESS PLAY ON TAPE prompt, go to File > Datasette Controls > Start. The tape then 'loads'.

In the Tape field along the bottom edge of *VICE,* you can find shortcuts to attaching tape images along with start/stop/rewind controls. Right-clicking this area displays a menu of programs on the virtual tape file that you can select, and *VICE* attempts to load them.

Alternative software
This guide has focused on using the *C64tapedecode* utilities to convert WAV files of C64 cassettes to the virtual tape TAP format. However, other tools are also available. From researching the options in preparation for this, the best alternatives are *TAPir* (**https://bit.ly/LXF315-TAPir**) and *C64tapwav* (**https://bit.ly/LXF315-c64tapwav**). As ever, we have relied on a tool that is relatively up to date; similarly, these alternatives have been updated within the last 10 years.

But you may well do your own research in this area and choose a different tool (or develop your own). TAP files are considered too big and slow, whereas PRG and T64 files are more lightweight and faster.

Unfortunately, as this is quite a niche area for Linux software, regular updates and new tools don't seem to be possible. On the other hand, *VICE* is regularly updated, so perhaps it's simply the case that these tools are pretty good at what they do. **LXF**

The Commodore 1530 C2N Datasette has an unusual connector, which means you need a dedicated adaptor for PC compatibility.

QUICK TIP

Don't confuse a TrueTape64–style device for a TapeCart or an SD2IEC. These are very different; the TapeCart emulates a C2N Datasette, while the SD2IEC is disk drive emulator. Both load disk or tape images from an SD card into a working Commodore 64, and often appear in eBay searches for TrueTape64–style devices.
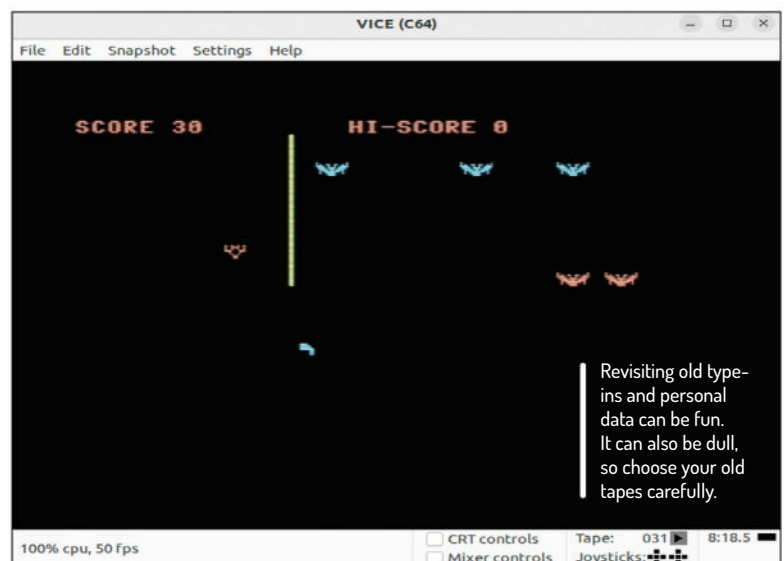
Revisiting old type-ins and personal data can be fun. It can also be dull, so choose your old tapes carefully.

» GET OUR RETRO MEDIUM: PAPER Subscribe now at **http://bit.ly/LinuxFormat**

AUDACITY

Credit: www.audacityteam.org

Enhance your system audio with USB power

Whether it's for podcasting, music production or simple day-to-day use, **Michael Reed** says a USB audio adaptor can greatly improve your sound.

OUR EXPERT

Michael Reed is a sound man in every sense of the term.

USB audio adaptors are external boxes with audio inputs and outputs that can be used in place of a soundcard or the built-in sound facilities of a PC. They usually offer a considerable increase in quality over the stock audio hardware. To add to this, some audio adaptors offer a greater number of inputs and outputs, specialist connectors for good quality audio sources such as microphones, and a higher recording resolution. Best of all, basic models aren't very expensive.

All laptop and desktop PCs have sound hardware of some sort. A desktop computer usually comes with line level inputs and outputs, and possibly headphone and speaker outputs. The quality of built-in audio is often passable, but the inputs tend to be subpar because few people use them and they're designed for telephony-type use, such as video chat, where quality isn't important.

Once you have your USB audio adaptor connected, it can take over for typical computer audio use, such as when watching streaming video sites or playing games, but with improved sound quality. These adaptors really come into their own, in terms of quality and ease of use, if you produce content involving audio, such as video blogs, podcasts or music.

Inside specifications

The most obvious starting point when choosing an audio adaptor is to consider how many inputs and outputs you need. A two-input, two-output device is capable of recording and playing back a stereo signal or addressing the inputs as two monophonic sound channels. These two inputs could be the input from a guitar and the input from a microphone placed in front of a singer. It could just as easily be the contribution of two separate podcast users.

You can get audio adaptors with more than two inputs and outputs. For example, you might want a device that could support four microphones at once for more advanced podcast recording. Or you might need
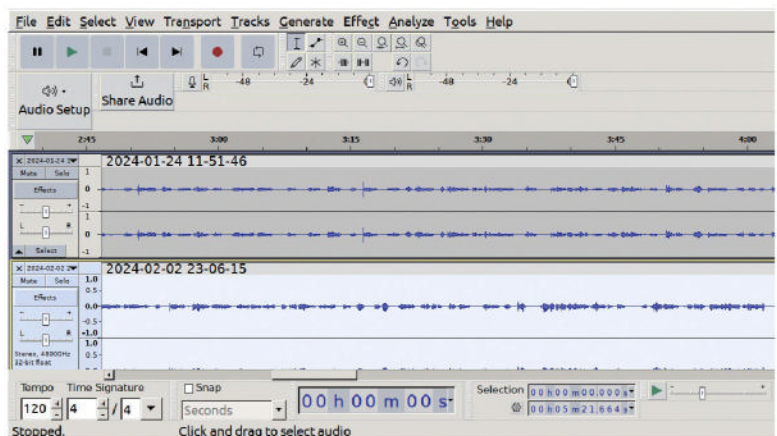
Audacity offers audio recording at perfect digital quality, even handling multiple, layered tracks.

an adaptor with enough inputs to record an entire band on their own separate tracks.

Be aware that not all adaptors with more than two inputs and outputs can record them separately. Many budget two-input, two-output audio adaptors have two microphone inputs and two line level inputs (four total), but they can only record two inputs at once.

Generally, a hobbyist, solo musician can get by with a two or four-input audio adaptor. Don't pay for inputs you're not going to use is our advice.

Audio input types

The standard audio inputs around the back of a PC tend to use 3.5mm (1/8-inch) stereo jacks, but these aren't very popular on external sound interfaces as they're not terribly robust, particularly if you're plugging and unplugging things all day long. Expect to find larger 6.35mm (1/4-inch) inputs, and these are generally monophonic rather than stereo. Such connections make more sense if you plan to plug different things into the inputs simultaneously. For instance, this happens to be the sort of connector that electric guitars and other instruments use. The XLR connector is another common sight on external audio interfaces, and it can accept a professional three-prong mic cable.

As well as physical differences between various types of connector, there are electrical differences. For example, the output from an instrument such as an

QUICK TIP

XLR microphone cables are balanced – negative and positive versions of the signal are sent through the wires. Any differences must have come from the outside of the cable and are removed electrically.

electric guitar is dissimilar to that of a microphone or a line level signal. If you think you will use your adaptor in this way, consider getting one with an instrument input, as this sorts out all the impedance and level matching for you. With a setup like this, you don't even need a guitar amp, as you can simulate that in software.

You might be able to plug your mic into a line level input, but you wouldn't be able to hear anything. If you want to use a mic, make sure your adaptor has a microphone input. Mics tend to output a minuscule signal that needs a lot of boosting, and this circuit is called a preamp. A switch on the adaptor that changes between line, instrument and microphone is common.

Some (not all) microphones that use an XLR connector need power to work. These tend to be the more studio-orientated musician and podcasting condenser mics. If you are planning on using a mic of this sort, make sure the adaptor can supply phantom power. This is a 48V supply that is transmitted through the same wires as the audio signal, and it's usually switchable between on and off on the front panel of the audio adaptor. Watch out for USB audio adaptors that have the XLR connector but no phantom power.

Maximum output

When it comes to outputs, things are simpler. Most basic adaptors have stereo line level outputs. These are perfect for outputting to powered speakers or an amp. A lot of units also have a headphone output, and it's worth going for one with its own volume control. Once you've got used to having a manual volume control in easy reach, you'll wonder how you managed without it, particularly if you use headphones a lot.

This brings up another point: controls. If it's the external box type of audio unit, it's common to have a combination of sliders, rotary controls and switches. These usually allow control over input volume, which is useful as various devices, instruments and mics have wildly different output levels. It's also important because most external USB audio adaptors have more gain than the normal inputs on a computer as they are designed to work with a wider range of input levels.

Mind your clippings

With digital audio, the loudest possible sound is labelled 0dB (0 decibels). In this digital world that we all operate in now, making sure that you avoid clipping (exceeding 0dB) is even more important because once

This recording of a human voice is clipped as it exceeds the maximum volume of the input.
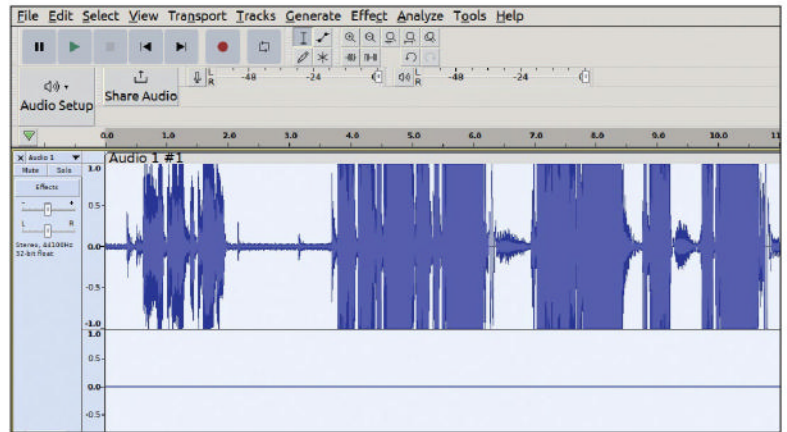
something goes off the top, things start to sound awful and there's no way of getting the lost information back.

You can use hardware or software metering to determine how close the maximum signal volume is to 0dB. Some audio adaptors have an LED meter, an example of hardware metering. The most basic is just a clip light that shows you when the signal is exceeding the maximum level, but a proper meter, that gives a full view of how close you are to maximum, is better.

Recording software, generally, offers an indication of the input level. The problem with software metering is that once the signal has been clipped at hardware level, there is nothing you can do to repair the damage.

Configure Pulse

You've got the USB audio adaptor and plugged it in – what next? First make sure that your normal software is using the adaptor as your primary sound device. Most Linux distros run PulseAudio, a system that allows more than one app to use your audio device at once. Usually, you can find the PulseAudio settings by clicking on the main speaker icon of your desktop environment and selecting Audio Mixer. Failing that, search within your application launcher for 'pulseaudio' or run `pavucontrol` in a terminal.

You might encounter a slight problem when selecting your new audio device because it isn't always
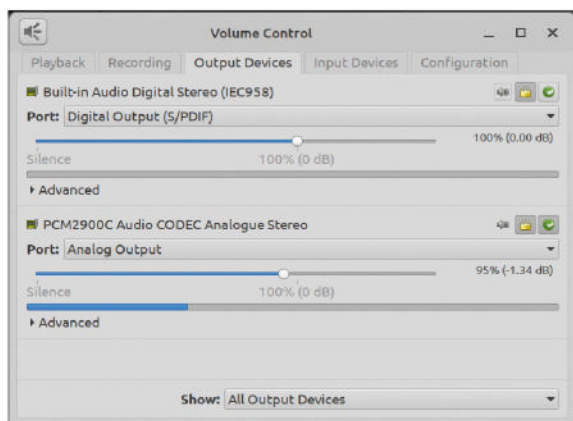
The PulseAudio setup utility. Make sure that the inputs and outputs are pointed to your external audio adaptor.

» CHOOSING THE ADAPTOR

The two main factors to consider are your budget and the features you require. There are audio adaptors, designed for use in recording studios, that cost thousands of pounds, but any of the audio adaptors from a reputable manufacturer are noticeably better quality than the built-in audio of a typical computer. The good news is that these basic models start at around £40-£50. That's amazing value for excellent quality audio when you compare it to other computer add-ons, such as faster CPUs or high-end graphics cards.

Linux compatibility is usually good for this class of device and nearly every recent model works with Linux once plugged in. Having said that, we still recommend an online search with the model name of the device you're thinking of purchasing, just to make sure other Linux users haven't run into any weird problems with it. If you use an online retailer, you can often search user reviews for the word Linux.

We definitely recommend getting a model with 24-bit recording and with a mixture of instrument, line level and microphone inputs. With that, you should be able to tackle most audio recording work.

obvious what it is called within the system. On a standard PC setup, your hardware might offer you three audio devices: the built-in audio, your graphics card's audio device (for HDMI audio) and your external USB audio adaptor. One way to find out what your USB audio device is called on your system is to issue the `lsusb` command at the terminal. For example, our M-Audio M-Track Duo device was listed as `Texas Instruments PCM2900C Audio CODEC`, and the `PCM2900C` part of the name was used by PulseAudio's configuration tools.

Other Linux audio tools use the ALSA name for the device, and you can list all detected ALSA devices by issuing the `arecord -l` command at the terminal. Between these two tools, you should be able to work out what your device is called on the system.

As for why this awkwardness exists, bear in mind that Linux has an advantage in that it's not normally necessary to install a driver for an audio device (and all the 'value added' cruft that normally comes with a Windows driver). The downside is that you have to rely on generic driver support for the chipset of the device. Similar detective work might be required to learn the names used for the inputs and outputs of your sound device.

In the PulseAudio (see – **LXF199**) Configuration tab, you can set a profile of Off for devices other than your USB audio adaptor. In the Input Devices And Output

Devices tab, select the Set As Fallback icon for your adaptor. This makes it the default audio device for all games and desktop apps.

To test that everything's working, run a desktop application such as a web browser to make sure that the sound is coming out of the audio outputs of your USB audio adaptor.

The Audacity of it!

Linux has a lot of powerful open source audio software available to it, along with some commercial applications that are worth considering. We'll start with one of the best open source audio programs, *Audacity,* for these examples. *Audacity* (see – **LXF297**) is a multitrack sound editor that can be used for recording projects such as podcasts or archival projects such as backing up audio from cassette tapes. It's not a dedicated music creation program, but it can be used for recording performances of solo artists or bands.

As *Audacity* computes all audio operations at an extremely high precision, once you have audio recorded into the program, there should be no further degradation of quality while you are working on it. If you have a good audio adaptor, you can start with a high quality input. Once you've recorded your audio, the audio adaptor you're using doesn't affect the audio fidelity per se, but having excellent sound quality means that you can be more precise in your sound editing and processing decisions.

At time of writing, *Audacity* has just transitioned from the 2.x series to the 3.x. We've found that, so far, 3.x is perfectly reliable and contains some tasty features that you'd probably be interested in, and we recommend that you go with that series. However, most of the distributions are still stuck on version 2.x for various reasons. According to the website (**www.audacityteam.org**), the recommended installation option for Linux is AppImage. The only downside of using that is that there is (by design) no system integration, because you don't actually install the package when using AppImage. The other options for getting the latest version are Snap, Flatpak and building from source.

Recording in Audacity

This isn't a full tutorial on using *Audacity*, but we'll give you the basic information you need to make a recording and play it back to test the system. We'll assume that you're using the *Audacity 3.x* series, and we'll make a stereo recording to keep things simple. Hopefully, as you have configured PulseAudio as the default audio adaptor, *Audacity* won't require any further configuration. If you do have difficulties with regard to making the inputs and outputs work, click the audio setup icon. Default should work, but if not, select your USB audio adaptor (having discovered the name of the device using the method we described earlier on).

Plug an audio source into the inputs of your audio device. This could be a microphone, for example. If so, connect the XLR cable to the microphone and plug the other end into the front of your audio adaptor. A word of caution: a microphone that doesn't require phantom power may be able to survive being fed it, if it's wired properly, but it's not advisable because it could damage

>> THE LINUX SOUND STANDARDS

ALSA (Advanced Linux Sound Architecture – **LXF108**) is the sound layer that actually interfaces with your sound hardware. It does this thanks to a lower layer of kernel drivers that talk to the hardware directly. Outside of highly unusual circumstances, you won't need to install any drivers to use a USB audio adaptor with Linux.

A snag with ALSA is that it can only handle one piece of software accessing a sound device at once. This is where the other software layers such as PulseAudio, JACK (See **LXF191**) and PipeWire come in. PulseAudio is the standard for typical desktop applications, such as web browsers and most games, that want to share the soundcard.

JACK (JACK Audio Connection Kit) is the musician's layer for sharing a soundcard. There are various GUI and text-mode apps to configure connections between music programs and sound devices.

PipeWire is the new standard that aims to replace both JACK and PulseAudio. If you have a modern distro, you're probably running it right now and it transparently sits between ALSA and PulseAudio and JACK for full compatibility with existing applications.
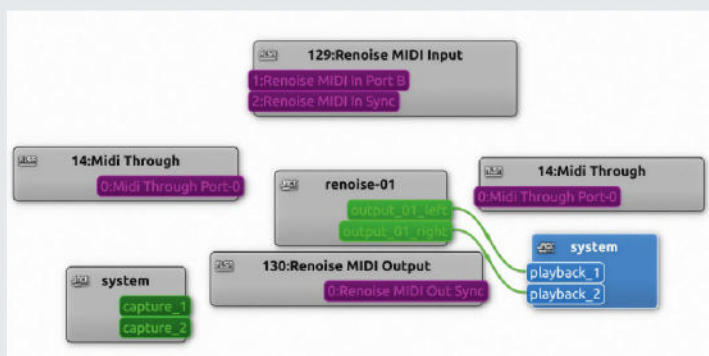
JACK is the musician's sound layer for controlling audio routing. These days, it's often running on top of PipeWire.

Ardour is the most highly featured open source music program on Linux. It can apply effects to incoming tracks without recording those effects.

it. Make absolutely sure that your microphone requires phantom power before enabling it on the adaptor.

Hardware versus software

Monitoring is listening to the thing you are recording while recording it. Be careful if you are monitoring using speakers as it's easy to set up a feedback loop, which sounds awful and can damage your speakers. What happens is that sound enters the microphone and is then played back through speakers. This sound then re-enters the microphone and then exits via the speakers again. Each time around, it gets louder and louder until you end up with a high-pitch howl. The other problem is that, if an open mic can pick up the sound from the speakers, it must affect the recorded audio to some extent. For this reason, we recommend headphones rather than speakers for monitoring.

We'll use our M-Audio M-Track Duo as an example. On the front, there is a switch that can move between three settings labelled Direct Mono, Direct Stereo and USB. Direct Stereo connects the inputs to the stereo outputs. This means you can hear the output of anything that is plugged into an input. In fact, we discovered that it works in this mode on a PC that is shut down (but powered). In this mode, you can use the controls of the device, such as the input volumes and the output volume, to control the level of the sound. This is a useful mode because it gives you practically delay-free monitoring of the incoming volume without having to set up any software at all.

Direct Mono is a variation on this that turns a stereo input into a mono one. If you plugged a microphone into input 1 on your device, the sound would only come through the left side of your headphones while monitoring. Selecting mono monitoring makes it so the sound comes through both sides, which is usually a more convenient way to work.

With USB selected, the connection between the inputs and outputs is cut. It is then up to the user to enable monitoring in the software that they are using. The advantage of software monitoring is that effects can be applied to the sound as it is being monitored. A singer might like to hear compression and reverb on their voice while singing, but the person making the

recording would prefer not to have those effects included on the recording. The downside is that there is a slight delay (latency) when using software monitoring.

Making a recording

We'll look at an example of recording some audio via a microphone. Adjust this example as appropriate for recording from other types of input. Make sure that your microphone is connected with appropriate settings on the front panel of your adaptor.

In *Audacity,* left-click on the microphone icon and select Enable Silent Monitoring. This causes the meter next to the microphone icon to show the current input level. Make some "teh, teh, teh, peh, peh, peh" type noise, as this contains what is called plosives, a part of speech or singing that's louder than other parts. Then try to speak or sing in a normal voice to make sure that you're getting the meter up to about halfway.

Having an adaptor with 24-bit recording helps here. The problem is that if you record at a low volume when using 16-bit recording (like most built-in audio hardware) and then scale the sound back up to full volume afterwards, it no longer uses the entire 16-bit range. You can get away with quite a lot with a 24-bit adaptor as it offers 16.7 million individual sound levels, as opposed to the mere 65,536 of 16-bit audio recording. To make things worse, internal audio inputs on a PC tend to add loads of electrical hiss to the sound, which gets amplified when you increase the volume of your recorded audio after the fact.

Once your levels are set, press the Record button. This gives you a display of the recorded signal as a continually expanding waveform in the main window. When you have your material recorded, play it back and listen to it.

Once you have determined that normal apps and games work properly, along with a simple recording and playback exercise, you have determined that your USB audio adaptor is fully working. Then it's time to start exploring the full world of audio-editing apps and plugins that Linux has to offer. LXF

All of these adaptors were immediately detected by Ubuntu Linux, and we've never had any problems with applications such as streaming video, gaming or audio recording.

Some Ansible advice for around the home!

Take work home? Not **Stuart Burns** – he applies home projects to work.

Of late, this writer has been busy creating new virtual hosts for different things and decided that it would be an ideal time to get stuck into *Ansible.* Things have evolved in system management since *Bash*.

Ansible is a great way to deploy software configurations to servers and Linux machines in general. (Windows works, too, but with caveats.) It's reliable, repeatable, easy to use and very extendable. And what you learn here can easily be expanded to work on remote machines.

We found ourselves building an *Ansible* configuration (aka a playbook) for our home desktops. That way, when the desktop needs rebuilding, it is as simple as a base install, install *Ansible* and run the playbook to put it back to just how you like it.

Ansible as a subject isn't served well by diving into every part of it straight away, but looking at it bit by bit, as it's needed, tends to work well. What follows is 'just enough *Ansible* to be useful' as an intro.

Why *Ansible*? Shell scripts still work but they can be clunky. Commands that would run to several lines in a shell script can replaced with just two lines in an *Ansible* playbook.

At the same time, *Ansible* offers consistency and is simple to understand. Sometimes *Bash* can become a bit gnarly.

Ansible playbooks can be rerun indefinitely and only changes are applied (or changes to put it back to how it was defined to look). *Ansible* uses a desired state to achieve this, so it's ideal for correcting configuration drift on servers.

Below is a quick but useful introduction to using *Ansible* for your desktop(s). First, you need to install it. On Ubuntu, for example, it's as easy as:

```
$ sudo yum install -y ansible
$ sudo ansible --version # Checking it's installed.
```

Ansible playbooks are YAML files that lay out desired configuration states. These are compared against what currently exists and the current state is changed by an *Ansible* task if needed to reflect the desired state. Each task is discrete and denoted as such in the YAML file. To try a trivial introduction, open a text editor and use the tasks below to perform an update and install some applications. In our example, the file is called **workstation.yaml**:

```
---
- hosts: localhost
- become: yes
- name: Update the OS
ansible.builtin.apt:
update_cache: yes

- name: Install apps
apt:
name:
- docker.io
- httpd
- vnstat
- python3
update_cache: yes
```

That YAML file can now be run using the command:

```
$ ansible-playbook -v workstation.yml
```

A quick tip here is to use one or more **-v** parameter switches to show what is happening as it runs its tasks. *Ansible* attempts an *Apt* update and installs those applications specified in the YAML file. It reports the output, similar to what is shown in the screenshot *(lower image, opposite)*.

Building upon that simple playbook, we can use *Ansible* to turn on and off services using the built-in tools. Turning on SSH could be done using a task that looks like this:

```
- name: Enable OpenSSH server
ansible.builtin.service:
name: sshd
enabled: yes
```

Hopefully, you can begin to understand that each action is achieved with a task. Each task has a name, a

» PRO TIPS!

YAML can be extremely picky – spaces, not tabs, and so on. To help mitigate this issue, we suggest creating the YAML code in an editor that both understands YAML and offers syntax highlighting. This helps reduce those annoying errors that manifest themselves in strange *Ansible* execution errors.

Here's another tip to avoid annoying 'What have I done wrong here?' scenarios. If there is nothing confidential in the YAML file, this checker can be very useful. If you are a bit confused as to why something isn't working, ChatGPT can fix up those errors in seconds. It is such an amazing time-saver. Just use "As an expert in *Ansible*, what is wrong with this code?" and paste in the code. It promptly highlights and rectifies the problem. Most of our issues seem to be indentation-related!

service to achieve the task and additional key pairs to provide more information.

The great thing about *Ansible* is that it is easy to do useful things quickly and learn as you go. One thing we did find ourselves doing was 'quick and dirty' *Sed* commands via the built-in shell command. They have their place but aren't what *Ansible* is about. Rather than teach bad habits, there are several ways to approach this problem.

Once you understand it, it does become trivial to tweak a file. The example below would look for 'foo' and replace with 'baa':

```
- name: Replace "foo" with "baa" in a file
replace:
path: /path/to/your/file
regexp: 'foo'
replace: 'baa'
```

There is an obvious need to be able to run external commands (but beware, *Ansible* can't monitor or fix what they do). To execute a task that runs a command can be specified as:

```
- name: Get network stats
ansible.builtin.shell: vnstat
```

Assuming the application is installed and within the path, it outputs network statistics.

In what we have done here, we have only touched the surface of what *Ansible* can do. We've tried to keep it as a simple but useful intro. Once you get used to how it works, you'll wonder why you ever bothered with complex *Bash* shell scripts.

A more in-depth look at what services can be used is officially documented at **https://bit.ly/3x3fiNH**. LXF

A snippet of the playbook that will be run later. Note the highlighting.

Running the playbook will give quite verbose output and report on what it did.

Undesired altered states

What do we mean when we say "desired state"?

We have spoken a lot about desired stated configurations. They are all about fashioning a system or set of systems into how the administrator wants that configuration to look. *Ansible* enables you to have groups of systems that share the same desired state.

Desired state does not mean everything on that machine – *Ansible* only cares about the items you specify. For example, it doesn't care if you installed some binary manually. In short, it's all about what's different since the last time the *Ansible* playbook was run.

That means the administrator needs to ensure that changes are performed only by *Ansible* to prevent a config drift that can be messy to sort out. While not an issue on a single machine, tens of machines could make for a bad day.

The great thing about a desired state is that if the playbook is run frequently – as an automated task, for example – it corrects any errant entries it sees. For example, if someone uninstalls a package mentioned in our playbook, rerunning the playbook detects that it is missing and reinstalls it.

Stuart Burns is a Linux administrator for a Fortune 500 company specialising in Linux.

» HERE'S TO THE NEXT 20 YEARS!

This month marks the 20th anniversary of Ubuntu. Mark Shuttleworth, its ultimate benefactor, claimed bug #1 to be Microsoft's dominance.

While that 'bug' is still true, Ubuntu has had an outsized effect on Linux for the mainstream, and a newer generation of Linux users got into it because of Ubuntu.

It took the great work of Debian and made it more accessible – it has better artwork, it's more consistent, and it's seen as the distro where stuff just works. It helped remove the stigma of 'doing trivial stuff shouldn't be this hard'.

Part of this was due to avoiding the 'no non-free firmware' dictum, thus allowing wireless and graphics to be less of an installation hurdle. Ease of use over licensing purity. That made a huge difference to those who see a computer as a thing to do work.

What really set it apart is that before fast broadband was a thing, Ubuntu sent out a free CD to anyone who requested one, with a cute little note asking people to pass it along to others after they had used it.

It's hard to overestimate what it achieved. Installing Red Hat in 1999 was tortuous. Five years later, Ubuntu just worked. It removed the 'this is too much like hard work' factor.

That's not to say that Ubuntu hasn't courted controversy. I was bemoaning this just two months ago. Ubuntu is now the 600lb gorilla of the Linux desktop and, as such, most other distros have to have feature parity with Ubuntu. It will be interesting to see what the next 20 years bring.

ONLYOFFICE

Credit: www.onlyoffice.com

Top AI helper tips for OnlyOffice

AI-based assistants can help create amazing office documents. The very real **Kseniya Fedoruk** guides you on using OnlyOffice's ChatGPT plugin.

OUR EXPERT

Kseniya Fedoruk is a document specialist at OnlyOffice and spends her days dreaming up documents and nights fighting AI-powered bots.

Artificial Intelligence (AI) is a buzzword that you hear everywhere. AI-powered technology is becoming part of daily life, changing our habits and the way we handle our everyday tasks and routines. Software development, content creation, data analysis, sales and marketing, and so on – all these fields and many others are affected by artificial intelligence, and its influence is becoming stronger.

Document editing is no exception. Nowadays, there are office software tools, including *OnlyOffice,* that provide AI-based features to make it easier for users to create, edit and collaborate on various types of content. In this article, we show you how to use *ChatGPT* within *OnlyOffice* to make document editing easier and more effective.

Activate ChatGPT

ChatGPT is integrated into *OnlyOffice* as a plugin that works in the background directly in the editors. To activate it, switch to the Plugins tab, launch the built-in Plugin Manager, and click Install next to the plugin.

Now click the Background Plugins button and start the plugin. Right-click at any point in your file and select ChatGPT > Settings in the context menu. Enter your OpenAI API key and click Save. The plugin is

No humans were hurt in the creation of this AI-generated text.
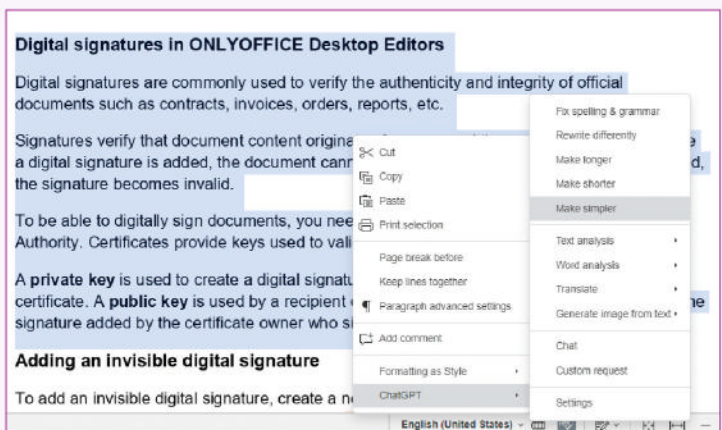
Available options of the ChatGPT plugin.

ready for work. You can use its features via the context menu. You can generate an API key in your OpenAI account even if you don't have a paid subscription.

Generate text

If you need to write something on a certain topic but have no idea how to begin, generate it to get you started. Later, you can use the AI-created text and revise it as you want. Let's imagine we need to write several lines about open source software. We don't have time to look up the required information online, so we ask the *ChatGPT* plugin to do this for us.

Launch the Custom Request window and type: "Please write 10 sentences about open source software".

Click Submit and wait for a few seconds while *ChatGPT* processes the request. You can see what we get as output in the screenshot *(left).* As you can see, we now have 10 sentences about open source software and can use this information for any purpose.

Along with the Custom Request option, you can chat with the AI assistant to ask any questions and find information, ask *ChatGPT* to tell you a joke and even generate simple code.

Add some style

Further options of the AI assistant when working with text include making it longer, shorter, simpler or even

www.linuxformat.com
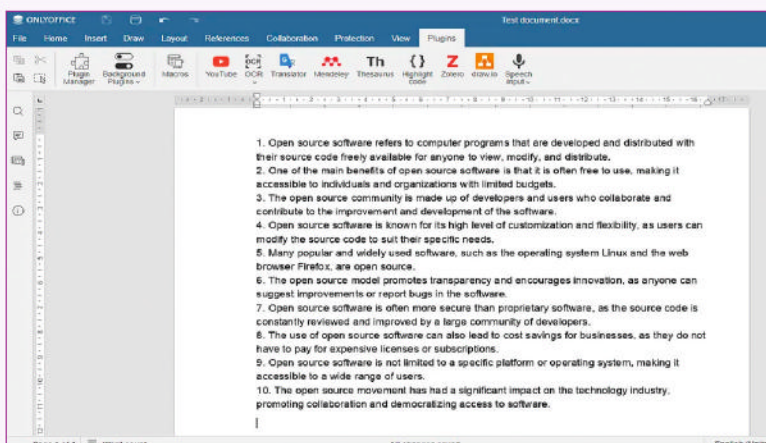
rewriting it completely. It's possible to use a combination of these options one by one to make any text look perfect.

Fix LXF grammar

With the *ChatGPT* plugin, there is no need to remember every grammar rule and know how to spell tricky words. The built-in spelling and grammar checking option enables you to worry less about how to write in the right manner, especially when you are typing in a hurry.

For instance, by clicking Fix Spelling & Grammar in the context menu to improve the text below, we were presented with a paragraph of corrected text, which you can see in the screenshot *(above-right):* The orignal looks like this: "Claud computin is provision of computin services including servers, storage, data bases, software, analitics, and inteligence, over Internet ("the claud") to ofer faster innovation, flexible resorces, and economys of scale. It allows users acces and use this services on demand, without the need for fysical infrastructure or teknical nowledge."
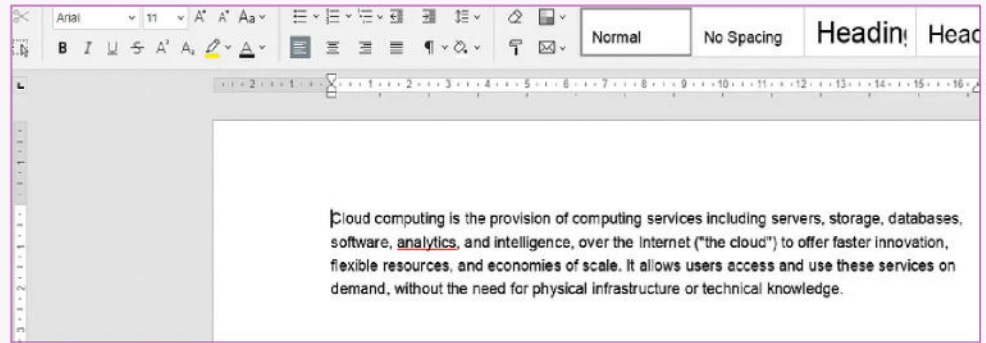
Analyse text

If you often deal with large texts, try the Text Analysis > Summarize option. For example, the chatbot can turn a 569-word piece of text about the history of the Linux operating system into a summary containing only 121 words. This way, you can quickly familiarise yourself with the most important information without having to absorb hundreds of words. Another option is the ability to extract keywords from the selected text: Text Analysis > Keywords.

Explain words

You can also use the plugin to explain professional notions and jargon for people who will read your document. When you choose the Word Analysis option from the context menu, you can select either Explain Text In Comment or Explain Text In Hyperlink.

The former option is intended to leave comments with explanatory notes, providing more information about the selected word or phrase directly in the file. When applying the latter option, the chatbot finds a suitable internet source and inserts the corresponding hyperlink into text.

Translate languages

When using the *ChatGPT* plugin, you don't need to use third-party translating services because the chatbot can translate chosen words, sentences or passages into multiple languages – just click the Translate option. Among the supported languages are English, Spanish, French, German, Italian, Korean, Chinese, Japanese and more.

Generate images

Another feature of the *ChatGPT* plugin available within *OnlyOffice* is the ability to generate custom images

based on your description. This way, you don't have to look for free pictures on the stock image websites.

For example, you need a title image for your essay or written project. What you want to see in your image is a smart-looking cat wearing sunglasses and sitting in front of a computer. Type what you imagine, click Generate Image From Text, and select the desired resolution of the output image (256x256, 512x512 or 1024x1024). The image-generation process usually takes a little longer than other requests.

Try ChatGPT

To test the *ChatGPT* plugin, you can quickly install *OnlyOffice Docs* or DocSpace on your own server. The recommended option is to use *Docker*. For *OnlyOffice Docs*, the *Docker* command is the following:

```
$ sudo docker run -i -t -d -p 80:80 --restart=always -e
JWT_SECRET=my_jwt_secret onlyoffice/
documentserver
```

Detailed steps on the installation of *OnlyOffice Docs* are covered in the official GitHub repo at **http://github.com/ONLYOFFICE/Docker-DocumentServer**.

For OnlyOffice DocSpace, the main commands are:

```
$ wget http://download.onlyoffice.com/docspace/
docspace-install.sh
$ bash docspace-install.sh docker
```

Find the complete instructions in **LXF314**, or create a free cloud account at **http://onlyoffice.com/docspace-registration.aspx**.

Please note that the OnlyOffice developers assume no responsibility for the accuracy or reliability of the information provided by *ChatGPT*. [LXF]

Our text after correction by the ChatGPT plugin.

» MORE ABOUT ONLYOFFICE

OnlyOffice DocSpace is a room-based environment for online document collaboration with colleagues, teammates, customers, partners and so on. The platform allows organising a clear file structure depending on individual needs or project goals. Flexible access permissions and user roles allow fine-tuning the access to the whole space or separate rooms.

OnlyOffice Docs is an open source office suite that comprises online viewers and collaborative editors for text docs, spreadsheets, presentations, forms and PDF. You can use it in OnlyOffice DocSpace or other cloud platforms, such as Confluence, Alfresco, Moodle, Jira, WordPress, Odoo, Redmine, Nextcloud and so on, as well as embed the editors into your own service/app and deliver it to end users.

» GET YOUR MONTHLY ADMIN FIX Subscribe now at http://bit.ly/LinuxFormat

The Epoch

It's Y2K all over again, as *Nate Drake* finds time to grapple with the impending Year 2038 problem.

On 2nd November 2000, a man calling himself John Titor began posting on Art Bell's BSS forums, claiming to be from the future year of 2036.

Titor painted a grim version of the 21st century, a cross between Pat Frank's 1959 post-apocalyptic epic *Alas Babylon* and the film *Civil War* that's currently hitting cinemas.

He predicted domestic uprisings in the US from 2005, culminating in World War 3 in 2015. In his future, government employees were being trained to use time travel to help mankind.

In 2000, Titor's wild predictions were unfalsifiable so provoked strong online debate. His story was lent a veneer of plausibility when he stated the original purpose of his mission: to travel to 1975 and retrieve an IBM 5100, one of the first portable computers.

At the time, it wasn't generally well known that certain versions of the 5100 had a secret function that could translate between the mainframe languages used by big infrastructure PCs and 'basic' computer languages.

'John' claimed he was selected for this mission because his grandfather had been involved in the

date, for example:

```
$ date -d @1710188659
```

Enter the superbug

The Year 2038 problem, also known as the Epochalypse, superbug or Y2K38, is similar to the Millennium Bug. It specifically affects systems that use Unix time, which includes major web servers, most programming languages and any Unix-based operating systems like Mac OS and Linux. Windows itself doesn't use Unix time internally but does so in various APIs and protocols.

It derives from the traditional use of a signed 32-bit integer to represent the **time_t** data type. Computer systems that do this can only store and represent **time_t** values of -2,147,483,647 to +-2,147,483,647.

In other words, on such systems, the earliest representable date is Friday 13th December 1901 and the latest representable date is Tuesday 19th January 2038.

One second after 03.14.07 UTC on 19th January 2038, this representation will be subject to roll over or 'overflow'. On unpatched systems, the **time_t** value

development of the 5100, which would make it easier to convince him he was from the future.

Though he never said so specifically, in the scenario John describes, the 5100 would be most useful for fixing the infrastructure of old computers to prevent the Epochalypse: a time computing problem that leaves some systems unable to represent times after 03.14.07 UTC on 19th January 2038.

State of the Unix

Unix time is measured from the 'Epoch' – midnight on 1st January 1970. This particular time and date was arbitrarily selected by Unix programmers, who required a uniform date for the start time.

On most platforms, Unix time is represented by a signed 32-bit integer detailing the number of non-leap seconds that have lapsed since the Epoch.

To see this Epoch time in any modern version of Linux, simply open the terminal and run:

```
$ date +%s
```

The terminal then displays the number of non-leap seconds since the Epoch in 1970. In our test, the output was 1,710,171,354, which is just over 54 years.

You can also use the *date* command in the terminal to convert the Unix time value to a human-readable

will then be set to 231 seconds before Epoch time (20.45.52 UTC on 13th December 1901).

Anyone who remembers the run-up to the Millennium with its dire predictions of planes falling from the sky, power outages and general hysteria can be forgiven for rolling their eyes, given how Y2K went largely without a hitch.

Although we feel this ignores the hard work done by programmers around the world to prevent this kind of Y2K scenario *(see boxout),* the Year 2038 problem isn't as easy to fix and the effects could be serious.

According to Jonathan Smith, a computer and Information science professor at the University of Pennsylvania, "Timers could stop working, scheduled reminders might not occur, scheduled updates or backups might not occur, billing intervals might not be calculated correctly."

It's also a concern for embedded systems, such as those used in aircraft and automobiles. Some of these, such as anti-lock braking systems, don't require dates, but others, like router firmware, need the correct time and date to function properly. Any software that uses X.509 certificates (like those used for HTTPS) and certificate authorities could also be affected, meaning secure web pages are inaccessible.

» COUNTDOWN Y2K

The Y2K bug centred on the fact that many programs stored year values as two digits rather than four. In unpatched systems, this could cause an erroneous year of 1900. It's easy to be dismissive, as it wasn't the apocalyptic event the media predicted, but this was thanks to coordinated efforts by programmers and business.

The issue dated back to the '60s when memory was rare and expensive, costing around the equivalent of around $10 per bit.

As the Millennium approached, more businesses started taking it seriously. By 1987, the New York Stock Exchange had hired a team of around 100 programmers to future-proof its systems against Y2K. In 1989, Norwegian programmer Erik Naggum was instrumental in updating RFC 1123 to use four-digit numbers to represent dates.

Still, it wasn't until the mid-'90s that programmers started pulling out all the stops. This was aided by the US government and its Year 2000 Information and Readiness Disclosure Act. It even formed a President's Council of senior officials from the administration and officials from agencies like the Federal Emergency Management Agency (FEMA), to monitor preparations by private companies.

The problem wasn't only software-based, as many hardware embedded systems, older mainframes and manufacturing systems also only stored dates as two digits. In the event, Y2K passed with only minor issues, leaving some media outlets to dismiss it as a hoax.

Best Buy may have overstated the Y2K problem by recommending shutting down devices, but the Year 2038 still lies ahead.

Psy's *Gangnam Style* became a harbinger of the Epochalypse in 2014 when YouTube's 32-bit counter was poised to break almost after viewing was set to go over 2,147,483,647 plays on 3rd December.

When YouTube was invented, the engineers simply couldn't conceive that a video could be watched that many times. In fairness, YouTube programmers issued a quick fix, Some applications that use future dates have already fallen foul of the Year 2038 problem itself.

For instance, 32-bit Android devices (v13 or older) will crash if the date's changed to 19th January 2038.

In 2016, a bug was also discovered in iPhones that would brick the device if the date is changed to the Epoch time of midnight 1st January 1970. A patch to fix this issue was released in iOS 9.3 later the same year.

The Epochalypse could also affect databases using 32-bit time fields, as well as database query languages (like SQL) with **UNIX_TIMESTAMP()**-esque commands.

Further proof that there's no quick fix emerged in 2006 with the implementation of AOL server software designed with a kludge of one billion seconds to make sure database requests wouldn't time out. Specifying an arbitrary time-out date 32 years in the future went beyond the Epochalypse date of 19th January 2038,

causing the time-out calculation to overflow and return a date in the past, causing the software to crash.

In the case of the AOL server, engineers simply altered their configuration file to set the database time-out rate to a lower value and all was well.

Still, proofing all databases, servers, programming languages and embedded systems against the Year 2038 problem may not be quite so simple. For instance, it's feasible to change the definition of the **time_t** data type in C to a 64-bit integer or to use units of 10 seconds instead of one. Coders could even deprecate **time_t** and replace it with a different value.

Epochalypse preppers

Programmatically-skilled readers will realise the dilemma with this: it would cause huge compatibility issues with legacy code. For instance, increasing the size of the **time_t** type to 64 bits in an existing system would cause incompatible changes to the layout of structures and the binary interface of functions. In other words, the attempts to fix the problem could cause the kind of disruption everyone wants to avoid.

The good news is that virtually all OSes now run on 64-bit hardware and use signed 64-bit **time_t** integers. This offers a wraparound date somewhere around the year 292,277,026,596, well beyond the retirement date of most coders today. Naturally, this works both ways – 64-bit signed integers can also represent times of up to around 292 billion years before Epoch time.

As for Linux, version 5.6 of the kernel and beyond supports a 64-bit **time_t** on both 32-bit and 64-bit architectures. As of kernel 5.10, the XFS filesystem supports inode timestamps and quota grace expiration timestamps also support dates up to the year 2486.

The ext4 filesystem, when used with inode sizes larger than 128 bytes, has an extra 32-bit field per timestamp, of which 30 bits are used for the nanoseconds part of the timestamp. Two bits are also sacrificed and added to the Epoch timestamp to extend its range to the year 2446.

Similar upgrades have been made to other Unix-based systems like FreeBSD and Mac OS to support 64-bit **time_t** on both 32-bit and 64-bit architectures.

As of *MySQL 8.0.28,* the functions **FROM_UNIXTIME()**, **UNIX_TIMESTAMP()** and **CONVERT_TZ()** can handle 64-bit values on any platforms that support them, including 64-bit Linux, Mac OS and Windows. Still, in relational database versions prior to August 2021, built-in functions like **UNIX_TIMESTAMP()** will return 0 after 03.14.07 UTC on 19th January 2038.

Overhauling **time_t** will solve the Year 2038 issue for executing most programs. But it doesn't resolve the issue of how to store date values within certain binary data files, many of which employ rigid storage formats.

There's also no easy solution for running 32-bit programs via compatibility layers or those that store time values using variables besides **time_t**.

In 2020, Debian developer Arnd Bergmann noted that full 64-bit time support would mean "around one third of our library packages would need rebuilding (and tracking) to make a (recursive) transition".

Futureproofing your system

As we've outlined, if you're running any version of the Linux kernel developed since 2020, it's likely your

system is fully Year 2038-proof. One easy way to double-check this is to open a terminal and request your system to display a date beyond the Epochalypse:

```
$ date -d "@2148485200"
```

If your system is Year 2038-proof, this command's output shows 12.26.40 EST on Saturday 30th January.

If you're coding apps, ensuring your OS and relevant libraries are up to date should ensure your programs are Epochalypse-proof. Different programming languages have taken varying approaches – for instance, as of version 1.9.2 of Ruby, the 2038 bug has been fixed by storing **Time** in a signed 63-bit integer, measured in nanoseconds since the Epoch.

An easy way to check if your chosen programming language won't throw up a horrendous error when dealing with dates beyond 2038 is to write a simple script to try to output one. For instance, in Python, first create a test script using a text editor like *Nano*:

```
$ nano test-2038.py
```

You can then paste in your test script:

```
import time
future_date = time.mktime(time.strptime("2040-01-01", "%Y-%m-%d"))
print(f"The future date in seconds since epoch: {future_date}")
```

If you execute the code via the terminal, you'll see an output listing seconds since the Epoch for that particular date (in this case, 1st January 2040).

Leaping into the future

So far, we've ignored the obvious elephant in the room: there are 86,400 seconds (60 × 60 × 24 = 86,400) in each day, so if we measure time from an arbitrary point like the Epoch date, it should be simple to calculate how many seconds have elapsed.

International Atomic Time (TAI) follows this model of exactly 86,4000 seconds to a day, but that doesn't exactly match the Earth's rotation around the sun (solar time), meaning it becomes less precise at the rate of around one second per year.

On a normal UTC (Coordinated Universal Time) day, Unix systems simply count from 23.59 on the previous day through to midnight at the start in a continuous way, measuring 86,400 seconds exactly.

However, to keep in sync with the Earth's rotation, solar time, UTC sometimes uses leap seconds, which occur once every few years. The IERS (International

Earth Rotation and Reference Systems Service) has inserted just under 30 such leap seconds since 1972.

As Unix time is measured at 86,400 seconds a day, this can cause discontinuity between it and UTC. Such systems square this circle by increasing Unix time into the next day during a leap second. At the end of the leap second, Unix time jumps back one second, returning to 24.00. For example, 1st January 1999 was a leap day, so as 31st December 1998 ended at midnight, Unix systems reset the time to 00.00.00 one second after midnight in order to stay in sync with UTC.

This means that the common belief that Unix time measures the exact number of seconds since Epoch time on midnight 1st January 1970 isn't quite true; you have to deduct any leap seconds to get the exact value.

Time to say goodbye

Given the preparations for the Year 2038 bug, it's unlikely to seriously impact critical infrastructure. Many 32-bit systems, even embedded ones, will have been worn out and replaced long before the Epochalypse.

Still, it seems programmers have learned the lessons of their forebears, as filesystems like ext4 have been future-proofed for four centuries. The updated **time_t** integer has a longer shelf life than the universe.

All the same, if John Titor's future employers are reading this, we can suggest a helpful alternative to a time-travel heist for an IBM 5100 to prevent the Year 2038 problem. The open source Hercules emulator (**www.hercules-390.eu**) has been available since 1999 and is compatible with all IBM mainframe operating systems, so can debug any leftover legacy code. **LXF**

Beloved by retro enthusiasts and time travellers alike, the IBM 5100 had sophisticated debugging features, supporting both APL and Basic.

» RETRO OVERFLOW

The 2038 bug is predicted on the fact that a signed 32-bit integer can only represent values up to 2,147,483,647. If a program tries to go beyond this, it will wrongly appear as a negative value, potentially causing it to crash. This is known as integer overflow and isn't specific to 32-bit systems.

One of the best-known examples is in the NES version of *Donkey Kong,* which uses an 8-bit value (maximum 256) to store the time left on any level. This is calculated by multiplying the level number by 10 and adding 40. If a player reaches level 22, the calculated value (260) 'wraps' around to 4, making it impossible to finish in the time available.

Integer overflows have their uses, too. For instance, the NES version of *Tetris* was designed to simply become progressively harder, with no end in sight. In 2023, 13-year-old Blue Scuti was effectively able to beat the game by hitting a kill screen caused by another overflow error.

Even consoles that use 32-bit signed integers sometimes hit on this hard limit. Players of *GTA V* are also limited to collecting a maximum of $2,147,483,647 for their criminal endeavours. If they try to pick up more money, their cash is wrapped around to a negative value, wiping out a player's wallet.

The PlayStation version of *Final Fantasy VII* has a damage overflow glitch in which a sufficiently powerful attack can create a negative damage number, defeating enemies automatically.

HotPicks

Darktable » GPU Screen Recorder » Ksnip » Koodo Reader » Ghostery » WeekToDo » Kodi » Tanks of Freedom II » Granatier » Czkawka » Encrypt0r

Mayank Sharma
can tell a beta release from a stable one just by swirling and whiffing a glassful of USB sticks. He picks another 11 fine vintage open source apps.

IMAGE EDITOR

Darktable

Version: 4.6.1
Web: www.darktable.org

I f you're a shutterbug, chances are you've got a camera that can save RAW images. And with *Darktable,* you can create a professional digital darkroom to convert and process your digital negatives.

The app is available as a Flatpak, so you can install it with `flatpak install flathub org.darktable.Darktable`.

Darktable loads straight into the interface, asking you to use the Import module to point it to some images. As with everything in *Darktable*, you get lots of options in the Import dialog as well.

First up, you need a mechanism to sort your images. You can do so by either rating the images, or colour coding them. You can then use the Sort By menu to sort images by rating or colour, which are just two of the 15 sorting options.

To run through the process a little faster, you can use keyboard shortcuts. Hold down the `h` key to view the shortcuts for popular actions.

To edit a photo, double-click it, which switches you to the darkroom view. This interface has the image in the centre and menus on either side, which contain everything you need to work on the image. These will make sense to someone used to the digital image-editing workflow. For everyone else, the app has an extensive multilingual user manual.

The good thing about *Darktable* is that it does non-destructive image editing. This means that instead of modifying the actual image while you edit it, the app records the changes in an XMP file, which are then applied to the image when you export it.

Once you've prepared the image, you can head back to the lighttable view and use the Export module. By default, files are saved on disk and as JPEGs. But you can choose to send the exported images to your Piwigo gallery, or as email attachments and more. Similarly, you can export them in any of the dozen image formats.

Darktable is just as good as, if not better than, other popular but proprietary (and expensive) image-editing apps designed for professionals.

LET'S EXPLORE DARKTABLE
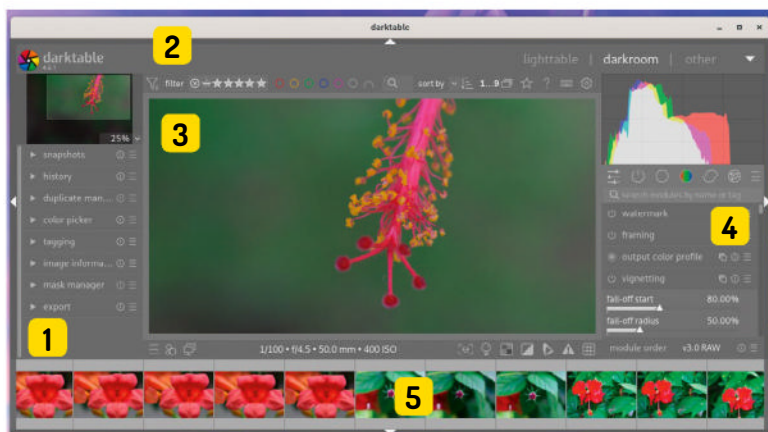
1 Left panel
You can use the modules in this area to find information about the image, take snapshots, export the image, and more.

2 Top panel
This area houses modules for common functions, such as the ability to filter and sort the images. It also gives you access to global settings and shortcuts.

3 Central area
The content here varies depending on the current view. It shows thumbnails in the lighttable view, and the selected image in the darkroom view.

4 Right panel
In addition to the histogram, this is where you can find all the modules that you need to edit your images.

5 Bottom panel
Another view-specific area, this area either shows a timeline of the image (lighttable view), or a filmstrip of images (darkroom view).

GPU Screen Recorder

Version: 3.3.0
Web: https://git.dec05eba.com/gpu-screen-recorder/about/

There are screen recorders aplenty for Linux, but what's different about *GPU Screen Recorder* is that it can go about its business without taxing your CPU. The app claims to be the fastest screen recording tool available for Linux, thanks to its minimal impact on system performance.

The good thing is that it works with both X11 and Wayland. It's available as a Flatpak, so you can install it on any distro with `flatpak install flathub com.dec05eba.gpu_screen_recorder`.

GPU Screen Recorder has a simple GUI with all its controls and settings neatly organised in a compact main window and it's pretty straightforward to use. You first need to choose whether you want to record a specific window, a follow-focus window or the entire desktop. Depending on your choice, you then pick the window, area size or monitor you wish to record.

Before recording, you can also tweak the frame rate and video quality. If you want to include audio

as well, just select your audio playback device from the drop-down menu.

From here you can stream the video to YouTube or Twitch, or record it to disk. The app supports three popular video codecs: H264, which is used by default on Intel cards; HEVV, which is used by default on both AMD and Nvidia; and AV1, but you won't be able to use it with Nvidia cards if you installed the app as a Flatpak.

For local recordings, you can select between four popular containers: MP4, FLV, MKV and MOV. If you know what you're doing, you can switch to the Advanced view and get more settings. For instance, by default, the app records videos in variable frame rate format, but you can switch it to using the constant frame rate instead in the Advanced view.

Besides streaming and recording, GPU Screen Recorder can also save Nvidia ShadowPlay-like instant replays of a specified duration.
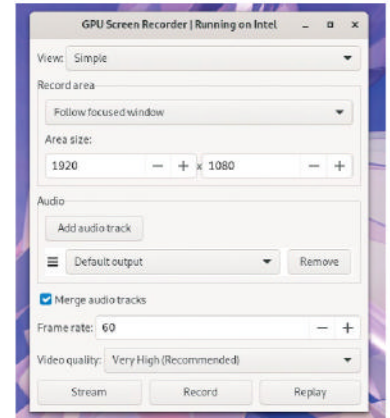
Ksnip

Version: 1.11.0
Web: https://github.com/ksnip/ksnip

Why do you need an app for taking a screenshot? Just hit PrntScr on the keyboard and you're done. That might do the trick for the majority of us most of the time, but there are situations when you need to edit that screenshot before you can use it. And *Ksnip* does exactly that.

Although *Ksnip* is a *Qt*-based tool, it works well with Gnome and atop both X11 and Wayland. The app offers precompiled binaries for RPM and DEB-based distros, and there's also the distro-agnostic AppImage.

To capture a screenshot, launch *Ksnip* and click on the downward arrow besides the New button. Select from one of the five listed options to select the area, window or screen you want to capture. The last option in the top dock lets you take a timed screenshot. Use the space to specify the capture delay in seconds.

Once a screenshot has been captured, *Ksnip* brings it up for editing. You can also head to File > Open to open an already saved screenshot.

Ksnip breaks up the editing and annotation options between two panels. In the vertical panel, you can find
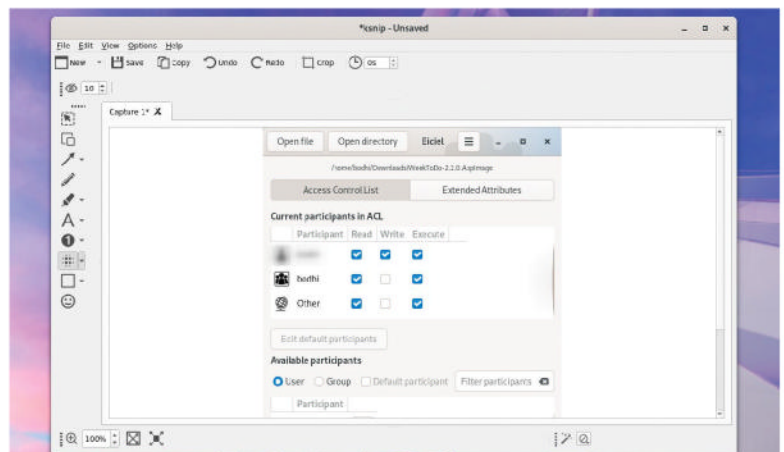
tools to blur or pixellate parts of the screenshot, draw lines, arrows or geometrical shapes, add text and numbers, with or without pointers, use a highlighter, and more.

The bottom panel, besides controls for zooming into screenshots, includes options for applying several image effects, such as a drop shadow, or add a border to the screenshot.

Once you are done editing the screenshot, head to File > Save to save the image automatically to the default location, which is your **home** folder. It automatically names the file with the **ksnip_** prefix followed by the current timestamp. You can also use the File > Save As option to choose the location and name of the screenshot before saving.

As well as saving screenshots locally, you can also ask Ksnip to upload them instantaneously to Imgur.

Koodo Reader

Version: 1.6.5
Web: https://koodo.960960.xyz/en

Almost all distros ship with a document reader that can open PDF files. However, most have a minimal feature set, like support for only a handful of file formats, or else a needlessly complicated interface. *Koodo* is a notable exception, in that it offers a clean, intuitive interface, and comes with extensive features and display options.

Koodo Reader isn't yet available in the repos of most mainstream distros, but the project offers precompiled DEB and RPM binaries, as well as AppImage and Snap packages.

The first order of business is to import books into the app. *Koodo Reader* supports all the popular ebook formats, including PDF, EPUB, Mobi, CBR and even DOC and DOCX. Use the Import button to point the software to the files you want to import. The app doesn't import books by folders, so you have to select them individually, though you can select multiple files in one go.

Once the files have been imported, you can catalogue them into shelves. By default, all imported
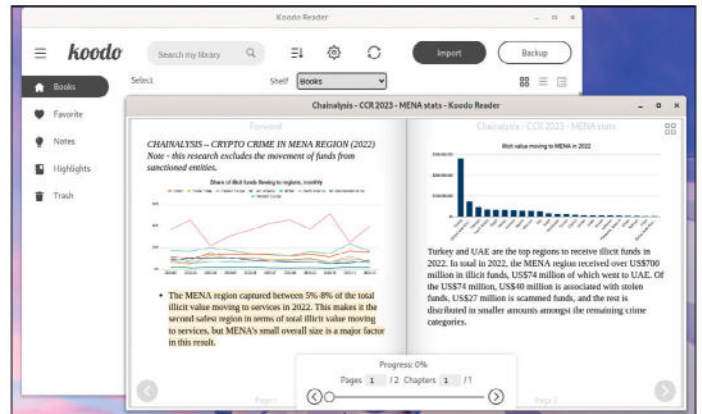
files are placed in the Books shelf. You can then right-click on a file and select the Add To Shelf option. Besides Books, the app has Study, Work and Entertainment shelves. You can also create new shelves from under the Add To Shelf option.

The best thing about the app is its reader view, which is stylised as a book, although you can switch to another layout. It offers controls to flip through the pages and bookmark a page, and even tracks time spent reading and more.

While it looks slick right out of the box, you can customise virtually every element as per your liking and improve your reading experience. You can change the font size, text colour, background colour, line spacing, overall brightness, paragraph spacing and plenty more.

With Koodo, you can highlight and annotate text, add bookmarks, and even add text notes to your ebooks.

Ghostery Private Browser

Version: 2024.03
Web: www.ghostery.com/ghostery-private-browser
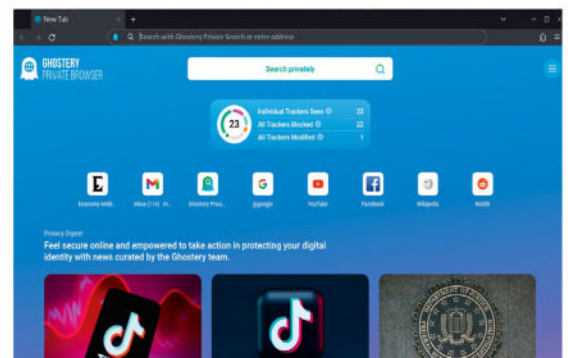
Brave is one of the best private browsers, but it isn't the only game in town. The appropriately named *Ghostery Private Browser* is another good option. Built atop *Firefox,* it uses the *Ghostery* ad/tracking-blocking extension to protect your privacy.

To install the browser, download the compressed tarball from its website:

```
$ tar zxvf Ghostery-2024.03.en-US.linux.tar.gz
$ cd Ghostery/
$ ./Ghostery
```

On first launch, it asks you to enable *Ghostery* to get started. Once done, you can use the browser like any other. It doesn't feel much different from *Firefox*, and can even install extensions from **addons.mozilla.org**.

The biggest difference, though, is that *Ghostery* blocks ads, trackers, cookies, pop-ups and a wide range of other forms of online tracking. And just like *Brave,* by blocking these invisible trackers, ads and pestering

pop-ups, the browser claims to save both bandwidth and time. The browser also has all *Firefox's* settings, such as telemetry, tuned for maximum protection.

Click on the *Ghostery* icon next to the address bar to see its magic. It displays the total number of trackers blocked, along with a breakdown of type, such as advertising, CDN, analytics and such.

In addition to blocking most trackers outright, the browser can also tweak certain ones. It does so in a bid to reduce site breakage and enable services to collect data, provided it does not compromise your privacy.

The landing page shows a summary of trackers encountered, blocked and modified. You can also tweak certain elements of its UI from the landing page. For instance, by default, it displays the search bar, stats, frequently visited websites, and news from the privacy digest. You can disable any or all of these from the settings offered in the landing page.

In addition to the in-app tweaks, Ghostery Private Browser defaults to its own homebrewed privacy-respecting web search engine.

WEEKLY PLANNER

WeekToDo

Version: 2.2.0
Web: https://weektodo.me

Most of the popular to-do list apps are designed for a team or angled at a professional setting. *WeekToDo* is one of the few exceptions. The app mixes the concept of a calendar and tasks list in a single interface. It offers just enough features to remain true to its name, which makes it an excellent planner for just about everyone.

The app offers precompiled binaries, as well as an AppImage and Snap packages.

WeekToDo takes you through a first-launch wizard. Besides routine tasks such as selecting the language and choosing a dark or a light theme, the wizard reminds you of its privacy promise, and how this means you'll have to use the app to export (and then import) your data if you need to switch computers.

The app offers all the basic features one might expect from a local weekly planner.

You can create tasks for every day of the week, just as you would in a calendar. You can also flesh out a task after creating it, and mark it as complete to strike it off your list.
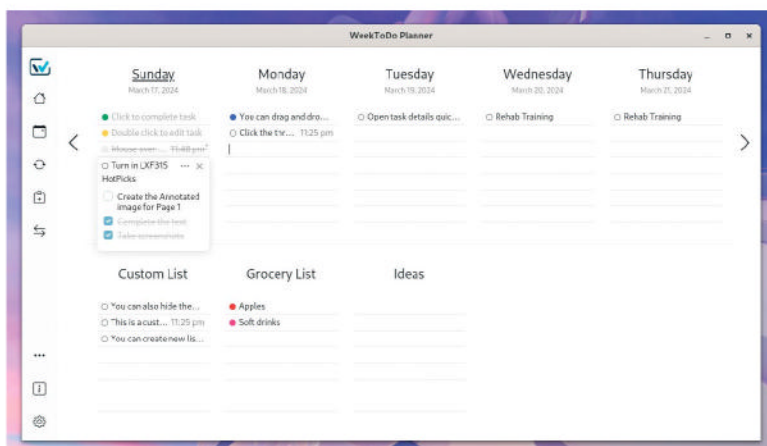
WeekToDo also enables you to create recurring tasks, which can repeat in one of the several available frequencies, such as daily, weekly, custom days of the week, and more. Tasks can also have completion times, and you can ask the app to notify you when the deadline approaches.

Tasks can also have subtasks, and you can drag and drop to move tasks around. And besides weekly tasks, you can also organise tasks and projects in custom to-do lists, which can also be reordered.

The best thing about the GUI is that it's quite customisable. For instance, you can hide various GUI elements such as the custom lists, and you can tweak everything from the size of fonts to how many entries are displayed per column.

WeekToDo is a wonderful cross-platform weekly planner that manages to tick all the right boxes for both form and function.

MEDIA CENTRE

Kodi

Version: 20.5
Web: https://kodi.tv

If you're looking to transform your spare computer into a home theatre, there's no better option than *Kodi.* Just hook up your big-screen TV and a hi-fi, and *Kodi* takes care of the rest.

The app is available on Flathub and you can install it on any distro with `flatpak install flathub tv.kodi.Kodi` .

One of the main reasons for *Kodi's* popularity is its ease of use and intuitive interface. The app has a 10-foot user interface that's designed for use with traditional televisions and can be navigated using a TV's remote control.

Very helpfully, the app ships with adequate defaults and only really needs to be pointed to your media library to get going.

Kodi helps you populate your libraries with content from your computer on first launch.

It supports all the popular audio, video and even image formats. Besides local content, you can also use *Kodi* to stream content from the web.

The app also has personal video recorder (PVR) functionality that can help you to record and watch live TV. *Kodi* supports all the popular PVR back-ends such as *MythTV, MediaPortal, Tvheadend* and *NextPVR*.
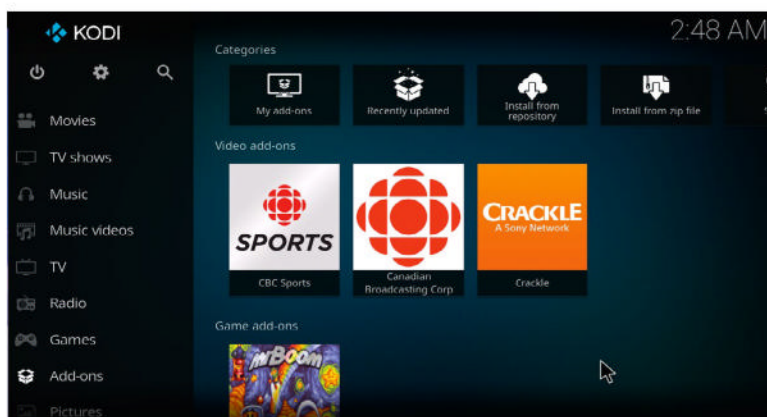
Kodi has an expansive list of features and offers a lot of conveniences for managing media. It has tools for managing libraries, creating and controlling playlists, and editing audio tags and other kinds of metadata as well. It also searches for subtitles and has extensive parental controls to restrict access to different areas of the app. And if its extensive list of features isn't enough for you, *Kodi* has a vibrant (*that's one way to put it!–Ed*) plugins ecosystem to enhance your viewing experience.

It's good the app ships with sensible defaults, because its extensive configuration options housed inside the unified settings interface can make the process of locating a particular setting cumbersome.

In addition to playing local content, Kodi comes with UPnP support that you can use to stream content to any UPnP-compatible devices.

Tanks of Freedom II

Version: 0.6.1
Web: https://czlowiekimadlo.itch.io/tanks-of-freedom-ii

T anks of Freedom II (ToF2) is a turn-based strategy game that you can either play against the computer or with up to three human opponents.

Download the zipball of the latest release from the game's website:

```
$ unzip tof-ii-* -d tof2
$ cd tof2
$ chmod +x ToFII.x86_64
$ ./ToFII.x86_64
```

This brings up the main menu, from where you can select a game mode. *ToF2* offers four single-player campaigns, as well as a skirmish mode, which can be played versus AI or against other players. There's a multiplayer option where you can go up against your friends on the local network. When you select this option, the game asks if you want to create a multiplayer session or join an existing one.

There's also the online option, in which you can go up against anyone on the internet. For that, the game asks you to first register your computer and get

yourself a player ID, which you can do from within the game itself.

There's a tutorial to acquaint new users, or you can just start with the prologue, which then unlocks the four campaign modes.

The objective in the skirmish games is to destroy all opposing nations by capturing their HQ buildings. On the other hand, campaign missions provide a greater variety of objectives, such as capturing specific buildings, defeating specific VIP units, escorting a VIP to a specific point, and more.

The game puts you in charge of an army of one of four nations. Each of these has access to the same set of buildings and units, differing only in visual style. Your task is to manage your resources and complete various objectives. Completed tasks earn you points, which you can then redeem to build new units, each having a specific use on the battlefield.

Tanks of Freedom II comes with three stock maps, and you can create your own using the custom map editor.

Granatier

Version: 1.4.24020
Web: https://apps.kde.org/granatier

G ranatier is a clone of the popular *Bomberman* game. Although included in the KDE games collection, *Granatier* is available as a Flatpak, so you can install it on any desktop with `flatpak install flathub org.kde.granatier`.

Gameplay involves strategically placing bombs, which explode in multiple directions after a certain amount of time, in order to destroy obstacles and kill enemies and other players in the vicinity.

The blocks hide bonuses and handicaps, which can either help or hinder your progress. Among the many bonuses that can be uncovered are those that make you move faster or increase your bomb's blast range.

But there are just as many handicaps, including those that slow you down or make you move too fast, or prevent you from placing bombs. Then there are a couple of ambivalent items, which can be both a boon or a bane.

The game offers two play modes. In one you go up against the computer, and the other is a multiplayer where you challenge other human players. Every round

you win earns you points, and the first player to earn a predefined number of points (the default is five) wins the game.

You can change the number of players, and their shortcut keys, in Settings. From under here you can also select the arena in which you want to play. Head to the Arena section, where you get dozens of choices, including the default Clanbomber arenas.

If you select Random Mode, a random arena is used for each new round. If there are arenas you don't want to play, you can untick them. And you can also create a custom arena.

To choose the players you want to play, go to the Player section and select the appropriate number.

Don't let its appearance fool you – Granatier has a simple objective, but going about completing it isn't as easy as it looks.

Czkawka

Version: 7.0.0 **Web:** https://github.com/qarmin/czkawka

A ll kinds of trash piles up in your distro over time. But the stuff that takes up the most space is the files that you forgot to get rid of after taking a backup. So, now you have two copies of the same file, just sitting there needlessly hoarding disk space.

Czkawka (Polish for hiccup) is a disk-degunking tool that can clean your disk of duplicates, temporary files, extra bulky files and a lot more.

The app is available on Flathub and can be installed with the following command: `flatpak install flathub com.github.qarmin.czkawka`.

At first glance, *Czkawka* doesn't appear very friendly, just like its name. On the left-hand side, you have a list of actions the app can perform, such as finding duplicate files, emptying directories, finding big files, similar images and video, and so on.

Depending on the option you select, the app displays relevant configuration options in the panel on the right. For instance, if you select Similar Images, you can pick the resize algorithm, the hash size and the hash type, with each setting offering several options in pull-down menus.
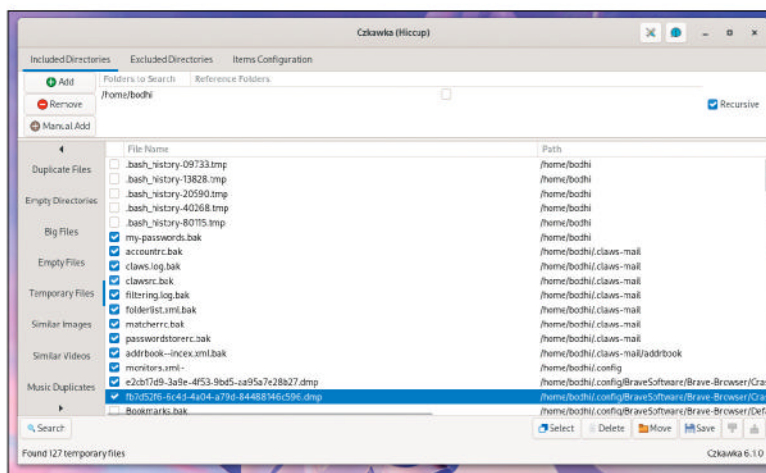
But unless you know what you're doing, you don't have to tinker with any of these options. The app does a nice job of selecting adequate defaults.

After you've selected the operation you want to conduct, point the app to the folder it should scan, at the top of the interface. By default, *Czkawka* looks inside your **home** folder, but you can change it to look inside a specific folder(s).

When done, smash the Search button and let the app do its magic. You can click on the results, which preview the image within the app. If they're the same, you can then choose to delete the duplicate.

Czkawka also has a CLI front-end, which is a handy option for terminal warriors who need to automate the degunking tasks.

Encrypt0r

Version: 3.12.13 **Web:** https://github.com/kunalnagar/encrypt0r

Y ou don't really need a reason to encrypt files, which is why there's a string of apps that'll help you with this. But *Encrypt0r* is a nifty little app that does little else.

The app does offer a DEB for installing the app on Debian-based distros such as Ubuntu, but it's best to use the precompiled binary. Grab the zipball from the website, as follows:

```
$ unzip encrypt0r-linux-x64.zip -d encrypt0r
$ cd encrypt0r/release-builds/encrypt0r-linux-x64/
$ ./encrypt0r
```

Encrypt0r doesn't have a fancy GUI, advanced features or annoying pop-ups. When launched, the app offers two options: Encrypt and Decrypt.

To encrypt a file, select the option from the main window, then point the app to a file, and enter a passphrase. The only option you get to load files is through a drag and drop. You don't get an option to browse your filesystem. You also can't drop multiple files at the same time, which makes sense, since you don't want multiple encrypted files to have the same password. But the inability to encrypt folders is a little irritating.

You also can't choose your encryption algorithm, and the app doesn't mention the one it uses. Before encrypting the file, the app prompts you for the location where you want to house the encrypted file. All encrypted files created with *Encrypt0r* have the .enc extension.

When you're done, hit the Reset button to return to the main menu.

To decrypt a file, select the Decrypt option. Now instead of a regular file, drag and drop an encrypted ENC file. Enter the passphrase and hit the Decrypt button. The app again asks you for the location where you want to store the unencrypted file before it unscrambles it. **LXF**

Encrypt0r is squarely aimed at new users who want to encrypt files without being bamboozled by a myriad of options.
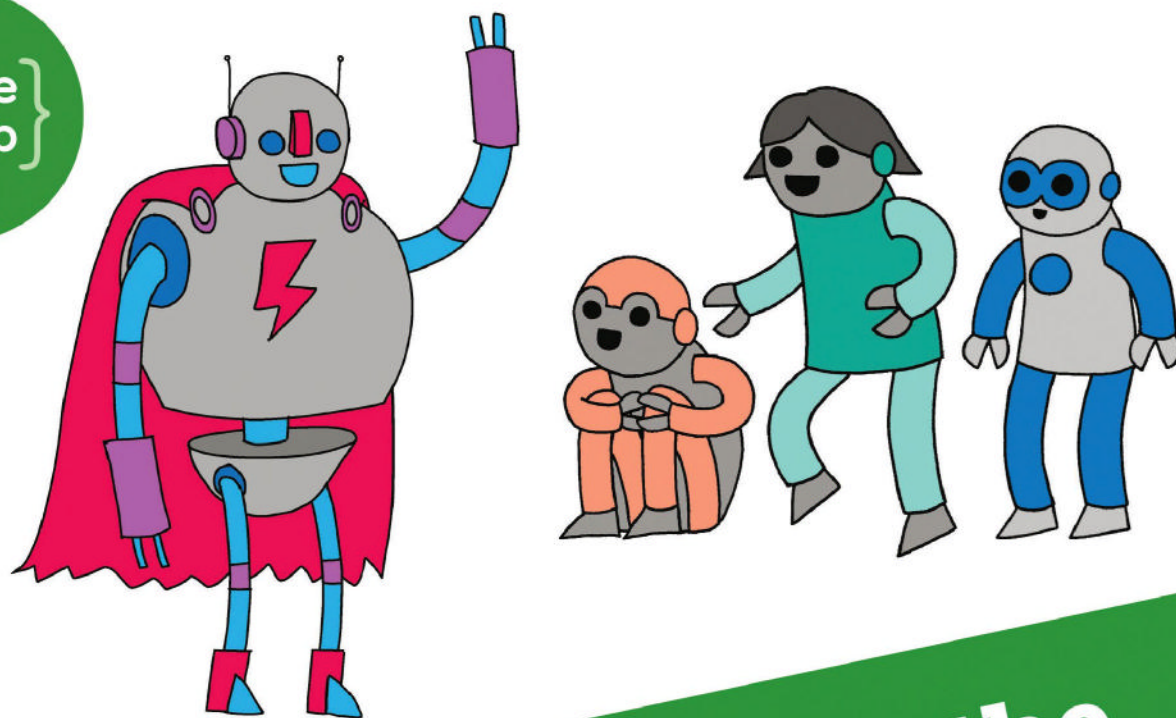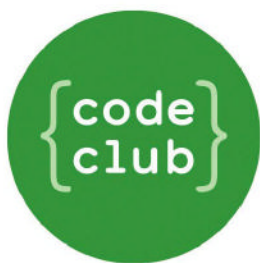
{code club}

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**

CODING ACADEMY

CLASSIC DEMOS

Let's code some demos like it's 1990!

Never let it be said that **Ferenc Deák** has his head in the clouds; more like in the past, as he explains how to recreate classic demo effects.

OUR EXPERT

Ferenc Deák never throws anything away, which is why he still has all his demoscene files. Sadly, it also means his fridge is a no-go zone.

I t all started when this author's beloved retro PC, used to host nostalgic gaming sessions, suddenly suffered a nervous breakdown and decided not to boot any more.

Only the noisy hard drive remained whirring away, when plugged into an adaptor, and burrowed in long-forgotten backup directories lay an intriguing folder called **scene**. Once-lost source files from a bygone era appeared that showcased how to create vibrant graphics, smooth scrolls and spirited sprite animations. Heavily influenced by the rediscovery of those files, the author has delved into his demoscene past, with the belief that such insights should not remain confined to the cylinders of a long-lost drive. The decision to share his knowledge with the world has led to the creation of this new tutorial series. So, let's code some demo effects like it's the '90s!

A diagram helps save a thousand words when talking of linear buffers.

QUICK TIP

You can find and clone all the code from the GitHub page: https://github.com/fritzone/lxf-demologia

Demo what?

The demoscene is a vibrant subculture that emerged in the realm of computer programming and digital artistry, characterised by the creation of real-time audio-visual demonstrations known as demos, as 'the kids' abhor long words. These demos showcase the technical prowess and creative talents of their creators, sometimes within strict size constraints, often pushing the limits of hardware capabilities to produce stunning audio-visual experiences.

Originating in the '80s Commodore cracking scene and gaining momentum in the '90s, the demoscene was fuelled by the desire of early computer enthusiasts to demonstrate their programming skills and artistic abilities. The primary motivation was to create impressive visual and auditory content that could run in real time on a variety of computing platforms, ranging from the Commodore 64, Amiga and Atari to, later on, even on contemporary hardware, such as PCs.

The demoscene community thrives on a culture of chivalry and rivalry, collaboration and respect, with individuals or groups, referred to as demo crews, mostly forming around one or more programmers (or coders, using the slang), coming together to produce demos that can compete in various demoscene

competitions. These competitions take place at so-called demoparties, events where demosceners gather to showcase their latest creations, share knowledge, engage in fiery but friendly competition and, last but not least, connect with like-minded individuals. Demoparties range in scale from small local happenings to large international events, such as The Gathering (Norway), Assembly (Finland) and Revision (Germany) – unsurprisingly, with thousands of participants.

The scene

From the C64, Atari and Amiga demos, which targeted specific platforms, to real-time programming under DOS, coupled with the coders pushing the limits of the increasingly powerful hardware of the '90s, the development of more intricate graphical effects became a trend in the demoscene.

While Linux adheres to the philosophy that everything is forbidden unless specifically allowed, older platforms ran along the lines that everything is possible, unless explicitly forbidden. It does not matter whether you want to set a weird screen resolution (can we use 256x256x256? Yes, we can!), or make your computer's speaker make noises like regurgitating yesterday's lunch, or – why not? – rewrite all the

characters so that you can have graphics running in text mode, nothing is impossible given enough time and dedication. The only drawbacks are that, well, sometimes you really can crash-freeze your computer, or if you are really unlucky, you might damage some hardware beyond the point of repair.

But those days are over, and we also have to evolve, otherwise we'll end up like the dinosaurs. It would certainly be easy to install *86Box* or *PCem* and a Turbo Pascal compiler, and just drop all the files into it, but we could count on one hand the people who would find it useful today. Nostalgic, yes, but not very practical.

So, in the name of pragmatism, in this mini series we are aiming to recreate and give a necessary theoretical background for some of the most notable effects from the demoscene of the '90s using C++ and SDL2. Please check out the boxout *(page 93) f*or a very brief introduction to SDL2, and we encourage our readers interested in this topic to read the SDL2 online (**www.libsdl.org**) documentation.

Effects of the '90s

In our context, an effect refers to a visually impressive and often innovative graphical element created by skilled coders, which are the building blocks of demos or intros. Effects can include sophisticated animations, intricate 3D graphics, complex shading, impressive transitions and various other experiments the coders can come up with.

However, behind almost every mesmerising effect in the demoscene there lies a foundation of complex mathematics. Whether it's the precise rendering of 3D shapes, the manipulation of colour gradients, or the seamless transitions between scenes, mathematics is the underlying force driving these digital marvels.

Our series kicks off with beginner-friendly yet captivating effects, ensuring that even those new to the field can dive right in. No need to fear complex mathematics and trigonometry yet. As we progress, we'll gradually delve into more advanced theories and corresponding effects. True to the tutorial's strapline, we're ultimately aiming to recreate the raw effects of the '90s, just before standardised interfaces and APIs, such as DirectX and OpenGL, and shader technologies became integral to the everyday life of the demoscene, because we do not wish to burden our readers with huge, cumbersome (albeit powerful) APIs. Let's stick to the basics, let's experiment and let's have fun while doing this.

The palette scroll

This is also known as colour cycling. In graphics programming, a palette refers to a set of colours that is used to render images or display graphics on a computer screen. The palette serves as a predefined selection of colours from which an image or graphic element can draw its colour information. Each entry in the palette corresponds to a specific colour, and graphics software references these entries to assign colours to individual pixels.

Mode 0x13 was the most popular graphics mode in DOS-based demos in the '90s. Having a 320x200 resolution boasting 256 colours, almost all the intros used this mode, because it was easy to set up and offered a 64,000-byte-long sequential memory, which

was directly mapped to the screen at the address 0xA000. In those days, this meant that if you wrote a byte to the address 0xA000:offset, the screen would be automatically updated.

In this mode, the palette consists of 256 entries, each representing a unique colour. Under DOS, the palette is typically manipulated using port I/O operations to set the red, green and blue (RGB) values for each colour entry. The RGB values determine the intensity of each colour component, allowing for a wide range of colours to be displayed on the screen. For example reddest red can be represented as the triplet (255, 0, 0), bluest blue as (0, 0, 255), and you easily can guess what greenest green is (0, 255, 0). All the other colours on a screen can be mixed from these three values.

≫ PARTY BACK TO '95

You have arrived at a party with your laptop in your bag. You are a lucky time traveller. Everyone else had to carry a 17-inch CRT screen and a beastly tower PC. It's time to dive in and find something fun to watch. At many parties, there are competitions that take place, each with its own unique characteristics. Most parties now host retro platforms, like Amiga, C64 or Atari, and if you want to stick to programming DOS, *DOSBox* is offered out of the box. Here's a breakdown of the most important types of competitions.

1. **Demo Compos** Impressive demos showcasing advanced graphics and sound. In the '90s there was no OpenGL, so each group had to write their own sound and graphics engine, but now parties support all the platforms one can write a program for.

2. **Intro (64k Intro)** Similar to Demo Compos but constrained to a smaller file size – maximum 65,536 bytes. Emphasises efficient use of space but mostly without music.

3. **4k Intro** A challenging category where participants create compact works, a maximum of 4,096 bytes long, the highest quality demos written in pure assembly.

4. **256byte Compo** As the name suggests, the biggest size of the demo can only be 256 bytes.

5. **RayTrace/GFX** A competition for high-res, high-colour-depth images, showcasing advanced graphical rendering techniques.

6. **Wild Compo** A competition with diverse content – films, photography – combined in the most creative way, not necessarily programming related.

7. **Lamer Compo** A humour-focused competition where the funniest creations win. Can be virtually anything that does not violate the code of conduct of the party.

8. **Muzax Compo** Involves the creation of 8 or 16-track music using MOD, S3M or XM formats, showcasing musical talent within specified constraints.

Demoparties are flexible, so everyone can find something, with no strict schedule. Pass time by hanging out with like-minded people, engage in gaming sessions, or relax by watching the amazing compos, which are recommended for everyone to attend at least one.
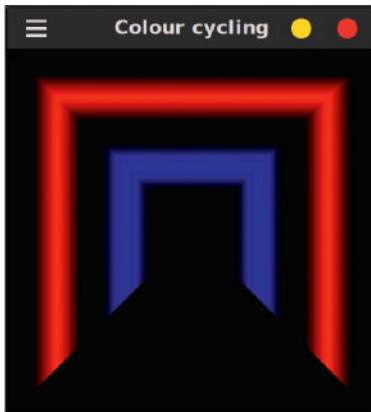
The fruits of our colour-cycling labour.

A very interesting effect that can be achieved without too much effort is if we draw lines on the screen, then create a black-only palette and insert a short gradient around a few indexes, then we rotate the palette. To uninitiated eyes, they will look like lines moving on the screen. Twenty-five years ago, colour cycling was easy – we just had to write out a bunch of numbers to a hardware port. However, under SDL2, we have to adapt to the possibilities offered by the framework.

The very simple palette can be created along the lines of the following C++ code, but please note that for brevity's sake, we've omitted the part that sets up the blue shade you can see on the screenshot *(above left)*.

```
void generateColorCyclePalette(SDL_Color* colours) {
  for(int i = 0; i < 256; i++) colours[i] = SDL_Color{0, 0, 0};
  for(int i = 0; i < 32; i++) colours[i + 32] = SDL_Color{i * 8, 0, 0};
  for(int i = 0; i < 32; i++) colours[i + 64] = SDL_Color{255 - i * 8, 0, 0};
}
```

The function initialises an array of **SDL_Color** structures to create a colour palette for the colour-cycle feature. The first 256 elements are set to black, followed by an increasing gradient of red shades from index 32 to 63, and a reversed red gradient from index 64 to 95. This results in a palette transitioning from black to red and back to black.

In order to faithfully emulate the demos of the '90s, creating an 8-bit graphics environment is crucial. In the modern multi-billion colour software/hardware environment, however, this can be challenging, so we have to resort to some exotic SDL2 features and create an 8-bit surface with the following function call:

```
SDL_Surface* surface = SDL_CreateRGBSurfaceWithFormat(0, SCREENSIZE_X, SCREENSIZE_Y, 8, SDL_PIXELFORMAT_INDEX8);
```

Although the call looks scary, there is nothing wrong with it: the first parameter is zero, because it is not used, then we specify the size of the screen – in the old days, this would have been 320x200; however, on modern computers, we can use any decent value:

```
const int SCREENSIZE_X = 512;
const int SCREENSIZE_Y = 512;
```

The **8** by itself specifies the depth of the surface in bits, thus we will have access to our required 256 colours. Then the real magic happens with `SDL_PIXELFORMAT_INDEX8`, which tells SDL to prepare a palette, with 256 colours, and attach it to the surface.

The palette that we have created a few lines above will be introduced to SDL via the call:

```
SDL_SetPaletteColors(surface->format->palette, colours, 0, 255);
```

What this short sequence of code does is the following: it sets the colours of the entire palette associated with the given surface to the values specified in the **colours** array, starting from index 0 to index 255. Right now, at this point, we have nothing drawn on the screen, just the palette, so it's time to put something on our (virtual) screen. The method **putPixel**

does exactly this:

```
void putPixel(int x, int y, Uint8 c, Uint8* screen) {
  screen[SCREENSIZE_X * y + x] = c;
}
```

The function **putPixel** is a simple routine for setting the colour of a pixel in a 1D array virtualising a 2D screen. The statement `screen[SCREENSIZE_X * y + x] = c;` calculates the index in the 1D array corresponding to the specified (x, y) coordinates and assigns the colour **c** to that index according to the explanatory image *(page 90)*, which shows how the index of a screen coordinate (x,y) is calculated when the memory is accessed in the form of a linear buffer.

Our linear buffer will be the following byte buffer, initialised to 0 by the `memset`:

```
Uint8* screen = new Uint8[SCREENSIZE_X * SCREENSIZE_Y];
memset(screen, 0, SCREENSIZE_X * SCREENSIZE_Y);
```

In the next step, we will render our drawings in the buffer we have just created using **initializeScreen**:

```
void initializeScreen(Uint8* screen) {
  for (int x = 0; x < SCREENSIZE_X; x++) {
    for (int y = x; y < SCREENSIZE_Y - x ; y++) {
      putPixel(x, y, x % 255, screen);
      putPixel(y, x, x % 255, screen);
      putPixel(SCREENSIZE_X - x, y, x % 255, screen);
    }
  }
}
```

This is where the fun starts, because here we can play with whatever contortion we want to generate, and in a few paragraphs' time we'll see another interesting graphic, too, but for this simple scenario, we have kept to rendering a few horizontal and vertical lines, that when applied to the palette scroll effect will give the illusion of walking through a square tunnel.

In the next step, we present that buffer to the screen in the main loop of the application. The SDL framework offers the possibility of rendering textures on to the screen via the use of the rendered surface. Sadly, hardware dropped support for creating 8-bit textures long ago, so here we also need to resort to some simple tricks – we will create a texture from the surface itself. And for those uninitiated in the dark slang of computer graphics, a texture is just a picture, which is worth a thousand words.

```
memcpy(surface->pixels, screen, SCREENSIZE_X * SCREENSIZE_Y);
texture = SDL_CreateTextureFromSurface(renderer, surface);
```
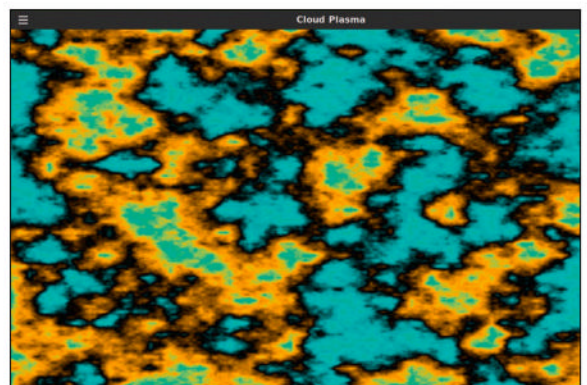

A classic-style plasma cloud colour cycle.

```
SDL_RenderCopy(renderer, texture, NULL, NULL);
SDL_RenderPresent(renderer);
SDL_DestroyTexture(texture);
```

Firstly, we copy into the pixel data of the SDL surface we have created, **surface->pixels**, the pixel data from the **screen** buffer we have rendered our lines on to, using **memcpy**. Subsequently, we create an SDL texture (**texture**) from the modified surface and render the entire texture on to the screen using the SDL rendering functions. As mentioned, support for 8-bit indexed textures is non-existent on modern hardware, so popular features, such as locking a texture and updating it, are not supported. This means that, unfortunately, we have to recreate the texture with every iteration. Thankfully, on our modern-day computers, this does not represent a challenge speed-wise, and offers the ideal opportunity for showcasing the effects we want to demonstrate.

The only thing that remains is the actual palette rotation and then the next iteration can set the palette to the rotated values via **SDL_SetPaletteColors**. This is done with the following piece of code:

```
void rotatePalette(SDL_Color arr[], int size) {
  SDL_Color c0 = arr[0];
  for (int i = 0; i< size - 1; i++) arr[i] = arr[i + 1];
  arr[size - 1] = c0;
}
```

The **rotatePalette** function simply rotates the given colour array, by saving the first element, shifting all the others to the left, then replacing the last one.

With all this in place, after building the application with `g++ colour_cycling.cpp -lSDL2 -o color_cycling` and executing it with `./colour_cycling`, we get a screen that looks like our screenshot *(opposite page, top-left)* while the bars move towards us, giving the illusion of moving through a square tunnel.

Plasma cloud fractal

Creating a square tunnel is adequate for an introduction, but we doubt it would place your demo high on the competition results list. If you want to get an effect that will really turn the eyes of your '90s mates towards your screen, you should try generating a cloud fractal and do colour cycling on that. The diamond-square algorithm is a method for generating fractal landscapes, commonly used to create realistic terrain or cloud patterns.

It involves initialising a 2D grid with corner values, then iteratively performing diamond and square steps. In the diamond step, the algorithm calculates the average of corner values in each square and adds a random displacement, forming a diamond pattern. The square step similarly calculates averages for diamond corners, adding random displacements to create square patterns. This process is repeated recursively, refining the terrain with decreasing random displacements until the desired level of detail is achieved, resulting in intricate and natural-looking fractal structures. The code for it is too long to present here, but there is a full, working example on our project's GitHub site *(see Quick Tip, page 91)*. To lure you into the trap, we've printed a screenshot of the

» IT'S SIMPLY A MEDIA LAYER

SDL2, which stands for Simple DirectMedia Layer 2, is a powerful cross-platform development library designed to provide low-level access to audio, keyboard, mouse and graphics functionalities. It is a popular choice for independent developers seeking a robust, platform-independent framework. While we won't go into details, here's an overview of how to initialise the SDL framework, and how to do some rudimentary drawing, by moving a square on the screen.

```
#include <SDL2/SDL.h>
int main(int argc, char* args[]) {
        SDL_Init(SDL_INIT_VIDEO);
    SDL_Window* window = SDL_CreateWindow("SDL2 Example",
SDL_WINDOWPOS_UNDEFINED, SDL_WINDOWPOS_UNDEFINED,
640, 480, SDL_WINDOW_SHOWN);
        SDL_Renderer* renderer = SDL_CreateRenderer(window, -1,
SDL_RENDERER_ACCELERATED);
        SDL_SetWindowFullscreen(window, SDL_WINDOW_
FULLSCREEN_DESKTOP);
        int xPos = 100, yPos = 100;
        bool quit = false;
        SDL_Event e;
        while (!quit) {        // Main event loop
        while (SDL_PollEvent(&e) != 0)if (e.type == SDL_QUIT) quit =
true;
        xPos += 2; yPos += 2;
        SDL_SetRenderDrawColor(renderer, 0, 0, 0, 255); // Clear the
renderer
        SDL_RenderClear(renderer);
        SDL_SetRenderDrawColor(renderer, 255, 0, 0, 255); // Draw
the rectangle
        SDL_Rect pixelRect = { xPos, yPos, 10, 10 };
        SDL_RenderFillRect(renderer, &pixelRect);
        SDL_RenderPresent(renderer);    // Put it on the screen
        SDL_Delay(16); // Adjust as needed for desired speed
        }
        SDL_DestroyRenderer(renderer);
        SDL_DestroyWindow(window);
        SDL_Quit();
}
```

This sets up the SDL environment via **SDL_Init**, creates the windows using **SDL_CreateWindow** and a renderer with **SDL_CreateRenderer** to actually be able to draw something. Just for fun, we make the window quasi-full-screen, and then repeatedly clear (**SDL_RenderClear**) and draw directly on the screen via **SDL_RenderFillRect**. That's it – no black magic!

application *(left)*. Now imagine all those colours moving hypnotically along the screen after you compile and execute the example from GitHub.

Stay tuned

This instalment of the series was pretty much static. We have drawn something on the screen and then, with smart colour manipulation, we gave the illusion of movement. But it's just an illusion. In reality, nothing moved on the screen, except the tiny red square flying outside of the box. Next month, we will bring some life to our static pixels, and make them move, dance, fly, morph and burn around the screen.

» IMPROVE YOUR CODE SKILLS Subscribe now at **http://bit.ly/LinuxFormat**

OSCAR64

How to compile C code on the C64

David Bolton takes you back to the 1980s and helps you develop Commodore 64 games faster by coding in C instead of 6502.

OUR EXPERT

David Bolton shows how Commodore 64 games and other programs can be written in C using the *Oscar64* C compiler.

Previously in *Linux Format...* We covered how to write 6502 (**LXF312**) first on a website and then using an assembler, and then how to develop for the Commodore 64 (**LXF313**). In this article, we're continuing with the C64 but moving way from 6502 assembly language and instead we're going with coding in C.

Originally, we were going to use the *CC65* C compiler but eventually chose an alternative, *Oscar64* (**https://github.com/drmortalwombat/oscar64**), instead. The main reason being that it appears to be more powerful, implementing most of C99 plus a lot of C++ features, and comes with a number of sample programs, including games. These include full source code, so if you want to write games or other programs, there are plenty of examples. The original idea for the *Oscar64* compiler was to translate the C source to an intermediate 16-bit byte code, with the option to use native machine code for crucial functions. This enables it do some extra optimisation.

Another benefit of coding in C is that the compiler includes code for doing multiplication and division. This isn't provided for 6502 assembly language, so if you need either, you'll have to write your own.

We'll look at a few examples in C, compiling them and running them on the *VICE* emulator.

Game elements

So, you want to develop a game for the Commodore 64? There are certain features that you need in every game. Here's a list:
- **Read the joystick**
- **Output text**
- **Manipulate hardware sprites**
- **Play sounds**

The easiest way to cover these is to take one of the sample games and see how it's done there. It's actually
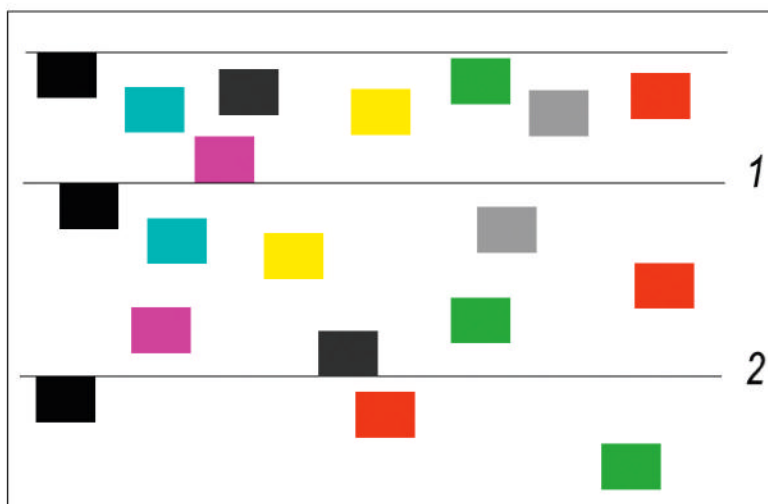
Nineteen multiplexed sprites on a Commodore 64 screen.

made easier because each of these has its own **.c** source file and header **.h** file.

The samples provided are very nicely structured and very well commented, making it fairly easy to understand the code.

The start of **missile.c** has these seven lines, plus a few words added to explain what each does:

#include <c64/vic.h> – defines enums plus various utility functions.

#include <c64/memmap.h> – some defines for char ROM (or not)

#include <c64/sprites.h> – functions for manipulating sprites.

#include <c64/joystick.h> – function for polling joystick

#include <c64/rasterirq.h> – raster interrupt functions

#include <c64/cia.h> – data for accessing parallel port (cia)

#include <gfx/mcbitmap.h> – functions for drawing, filling and so on, in hi-res mode

These refer to the files in the **Include** folder and specifically the C64. There are similar folders for Vic-20, C128, Nes and others. If you want to understand what the addresses mean, load the Project 64 text file (**https://bit.ly/lxf313map**), then you can search it.

Section of missile.c on the left and the code generated on the right.

```
1090 .s380:
1091 1174 : a9 02 __ LDA #$02
1092 1176 : 8d fd 41 STA $41fd ; (TheGame + 0)
1093 1179 : a9 00 __ LDA #$00
1094 117b : 8d 96 41 STA $4196 ; (CrossP + 0)
1095 117e : 20 82 22 JSR $2282 ; (screen_init.s1000 + 0)
1096 1181 : 20 56 32 JSR $3256 ; (missile_init.s0 + 0)
1097 1184 : 20 98 32 JSR $3298 ; (explosion_init.s0 + 0)
1098 1187 : 20 d6 32 JSR $32d6 ; (icbm_init.s0 + 0)
     118a : a9 1e __ LDA #$1e
1100 118c : 8d c0 44 STA $44c0 ; (nmissiles + 0)
1101 118f : a2 00 __ LDX #$00
1102 .l433:
1103 1191 : a9 5c __ LDA #$5c
1104 1193 : 9d 19 c8 STA $c819,x
1105 1196 : e8 __ __ INX
1106 1197 : 8a __ __ TXA
1107 1198 : 30 05 __ BMI $119f ; (main.s434 + 0)
```

Note than in **missile.c**, the game is played using a joystick, as are all of the other games. The keyboard is used far less and, in fact, the only sample that uses it is the paint program in the **hiresmc** folder.

A look at the hardware

Commodore produced a Programmer's Reference Guide to the Commodore 64 (**https://bit.ly/lxf315guide**). It's a great way to get to grips with it but it's not a quick read. Let's look closer at the C64 hardware. Although we're running an emulator *(VICE),* it mimics the hardware very closely.

The C64 has 64KB of RAM but parts of it are used for the character ROM, the char screen and the graphics screen. Effectively, this leaves you with 39KB for your code, data and so on.

The screen modes are either text or graphic. With text, it's 40x25 characters, while with graphic it's 320x200 pixels. But you can also vary these with multicolour pixels. One of the neat features of the C64

is that you can mix screen modes, and in **missile.c** this is done with the top of the screen in high-res mode and the bottom of the screen in text mode.

There are eight hardware sprites, each 24 bits wide by 21 rows high. These can be positioned anywhere on screen, including on top of each other. A very advanced technique is to have more than eight sprites, by repositioning sprites while the screen is being drawn. The sprites are sorted by vertical position and an interrupt can be set to go off at the start of certain raster lines, like those marked 1 and 2 *(see screenshot, opposite page).* The hardware sprites are then repositioned further down the screen, so you see 19 sprites in action each frame, so long as the repositioned sprites don't overlap vertically.

If you take a look at the header file **includes\sprites.h**, you'll notice that not only are there the real sprite functions, but there are also virtual sprite functions, letting you access up to 16 sprites through multiplexing. These have the same name as the real

> **QUICK TIP**
>
> Keep a map of all the locations your program uses and the name for them. The idea is to reduce aliases (multiple accesses of the same locations), because those prevent the C compiler being able to optimise.

>> GET STARTED WITH OSCAR64

Written in C++, *Oscar64* needs to be compiled. This is not difficult but the instructions provided on the web page are slightly incorrect.

First, you need to create a folder where you'll install *Oscar64*. Ours is **~/development/C**. Do a `cd` to there.

```
$ cd ~/development/C
```

Download the zip file from **https://github.com/drmortalwombat/oscar64**. Click on the green Code button at the top of the page and on the pop-up menu that appears, click Download ZIP at the bottom of the menu.

Once the ZIP is downloaded, right-click on the file, which should be **oscar64-main.zip**, and click Extract Here. This creates a folder called **oscar64-main**, and if you view it in *Files,* you'll see several subfolders within it, including **make** and **oscar64**. The source files are contained in **oscar64**. Now `cd` to the **make** folder:

```
$ cd ~/development/C/
oscar64-main/make
```

You should see a single file: **makefile**. To build the *Oscar64* executable, run the following command in the terminal. You need to have *g++* installed. Install it with:

```
$ sudo apt install g++
$ make -f makefile
```

That's all you need. Note: this is slightly different from the instructions on the web page. You get a few warnings, but after 30 seconds or so, you should see a **bin** folder under the main **oscar64** folder with the compiled executable file.

Next, you need to `cd` to the **games** folder under **oscar64-main/samples** and run **make.bat**:

◖ The VICE emulator running the missile.prg game.

```
$ cd ../ oscar64-main/samples/games
$ make.bat
```

This builds all seven games and generates a PRG file for each. Assuming you have *VICE* installed and opened *(see tutorial,* **LXF314**), just drag a PRG file on to it to start the game.

Demo program sprmux64 showing 64 on-screen sprites in a frame.

sprite functions but prefixed with a **v**. So **spr_image(** changes the image in a real sprite and **vspr_image(** does the same for a virtual sprite.

The number 16 is set in a **#define VSPRITES_MAX** (line 64), so it can be altered and *Oscar64* recompiled if you need more than 16 virtual sprites, but it still needs careful programming to avoid overlapping.

Reading the joystick

This is done through an interrupt that happens once a frame. It's set up in lines 844-848 of **missile.c** and this calls the function **joy_interrupt()**. Joystick values are read into the **joyx**, **joyy** and **joyb** (button press) variables. These are defined in the **joystick.h** file in the **includes** folder. There are two of each (0 and 1), supporting two joysticks, but in this case only joystick 0 is used. Line 8 in **joystick.h** shows where these joystick values originate: addresses **0xdc00** and **0xdc01**.

The joystick x/y values are used to move the crosshair sprite in line 569 by calling **spr_move()**.

The keyboard works in a similar way, except that it's not done in an interrupt. Just call **keyb_poll();** in a loop then do a switch on **keyb_key**. Lines 127-173 in **hiresmc\paint.c** show this in action.

Text screen modes

For this we can look at **games\lander.c**. This is one of the shorter games and is probably the best game to start with. It's less than 300 lines long and uses three sprites on a text screen. The text screen is what you see when the C64 boots up. It has 40x25 display that is mapped into memory at **0x0400** and for each character location, there is an equivalent colour map byte at **0xd800**. These are defined in **lander.c** at lines 13 and 14. So writing a coloured char to the screen is done in the function **screen_put** (lines 39-45).

```
define Screen ((byte *)0x0400)
#define Color ((byte *)0xd800)
```

Colours in the text mode are byte values 0-7. You can enable multicolour mode by using byte values 8-15.

Hi-res screens

You have full control of the screen with two **include** files: **mcbitmap.h**, which uses **bitmap.h**. Both libraries are found in the **gfx** folder. The **bitmap.h** has a number of functions for drawing shapes (circle, triangle, polygon), single pixels, lines, bitmap images and text characters. Both files use a combination of C and 6502 assembly language to draw the graphics fast.

Examples of this include **char_write()** (line 124 of **missile.c**), used to draw text such as 'Bonus' and 'Level'. It calls **char_put** for each character, which calls on **bmmc_put** (line 35 in **mcbitmap.c**) to do the actual drawing.

There are limitations – you get two colours in each 8x8 pixel area or you can enable multicolour mode (like text), but that reduces the horizontal resolution to 160 pixels wide. Explaining all the colour possibilities would take up a lot more space here. Pages 99-130 in the C64 Programmer's Reference Guide go into a lot more detail and are a real must-read to understand this.

Playing sounds

The Commodore 64 comes with a SID chip; this is a mini synthesiser. These are some of the features that it provides:

- Three tone generators (voices), frequency 0-4kHz (16-bit decomposition);
- Four forms of waves (sawtooth, triangle, rectangle pulse width modulation, (white) noise /rush);
- Three amplitude modulators, until 48dB;
- Three envelope generators;
- Synchronisation of the oscillators.

To use this in games, you need to include **sid.h**. This is used in the function **sound_trigger** in the **breakout.c** game lines 301-350 and called in various places in the game when the ball bounces off a side wall (line 268), top wall (line 374) and the bat (line 400), and hits a brick (line 453).

Embedding binary data

If you have graphics for sprites, bitmaps, music or sounds in a binary format, then you can include them directly in the code using the **#embed** directive. This can be seen in lines 14-17 shown below in **missile.c**:

```
// Sprite assets
const char MissileSprites[] = {
#embed "../resources/missilesprites.bin"
};
```

The **samples\resources** folder contains a number of binary files.

There is a load of useful information on the main page, including pragmas such as **_striped** for arrays and structs that alters the layout like this:

```
int array[8];
```

The memory layout of this array (remember ints are 16-bit in *Oscar*) is:

» C VERSUS 6502 ASSEMBLY – FIGHT!

Back in the '80s when home computers such as the Commodore 64 first appeared, Basic programs were just too slow. To get decent speed, you had to write games in assembly language: 6502 for the C64 and BBC Micro, and Z80 for the ZX Spectrum, Dragon and so on. But as people got into Amigas and Atari STs (both 68000), and then PCs in the late '80s and '90s, C compilers became available. C was invented in 1972 and is the only programming language to be found for every CPU. It has been described as a portable assembler and makes porting programs between different CPU types possible without a complete rewrite.

Also with C, programs are easier to understand. In 6502 assembler, you spend your time moving bytes between CPU registers and memory. In C, you can use structs, arrays and pointers. In the **missile.c** game, an **Explosion** is a small struct holding x and y coordinates and a radius and a pointer to the next explosion in the linked list.

When *Oscar64* compiles the C program, it generates an assembly language disassembly, so you can view the C and the code. In **missile.c** there are five lines of code in **explosion_init()** (line 214). This translates to about 35 lines of 6502 at line 5852 in **missile.asm**. Given that you write C and assembly language at the same number of lines per second, C is seven times faster to write.

```
LHLHLHLHLHLHLHLH
__striped int array[8];
LLLLLLLLHHHHHHHH
```

ZeroPage

By default, global variables are stored in main RAM, but **__zeropage** stores them in the area from **0x80..0xff**. This should only be done if your program isn't making system calls to the kernel:

```
__zeropage int a;
```

The C compiler has support for a number of pragmas that affect the code produced during compilation. There are pragmas for:

- Specifying heap and stack sizes;
- Compiling source files so no makefile is needed;
- Changing the character map;
- Specifying optimiser options;
- Unrolling loops for better performance, especially with scrolling;
- Marking functions to compile directly to 6502 machine code rather than going through the intermediate code stage first;
- Controlling where in memory code is generated for – data can be placed directly at addresses, so, for example, char maps don't need to be copied;
- Putting code in overlays that are loaded during program execution.

Embedded assembly

The C compiler can include 6502 code inside an **__asm** block. Labels are supported and return values can be done by storing values in page zero locations accu. Search for the word accu in **.h** and **.c** files to see examples. Here, for instance, is a C function to add two 16-bit ints:

```
int asum(int a, int b)
{
        return __asm
        {
                clc
                lda    a
                adc    b
                sta    accu
                lda    a + 1
                adc    b + 1
                sta    accu + 1
        };
}
```

Design tips

Using C on a 6502 involves a few compromises and you should bear these in mind when writing software. The 6502 does not have a large stack and the C compiler tries to analyse the call graph and then builds a static stack. Recursion does not play well with that! For the same reason, avoid function pointers and use switch statements instead.

Use unsigned arithmetic as it's quicker than signed arithmetic. Use **enums** instead of **#defines** for constant values. This lets the optimiser do a better job. Look at the various optimise compile options. Do you need to optimise for size or speed?

Use the **__assume** keyword for telling the compiler about things such as limited ranges, non null pointers and unreachable default cases in switch statements.

The *Oscar64* package is nicely put together and the sample programs including games show it off well. The C compiler also supports a number of C++ features, including namespaces, reference types, templates, vector, array and list template classes, lambda functions, for range loops, plus new and delete. **LXF**
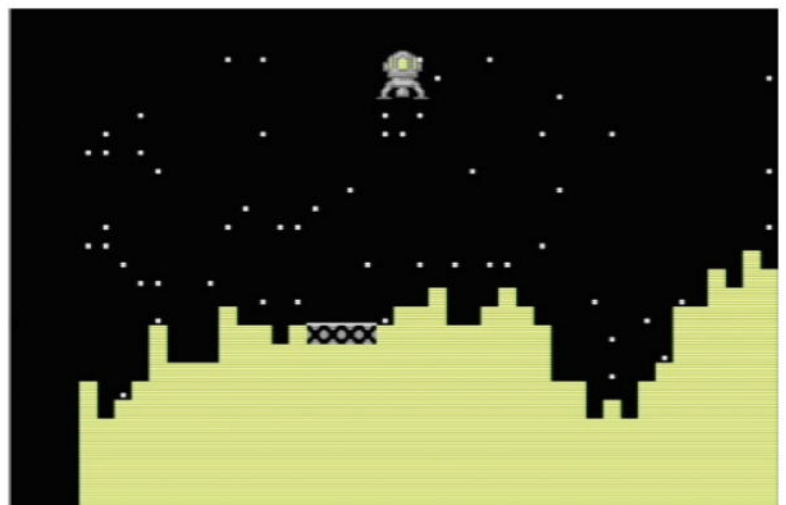
■ The Lander game compiled from lander.c.

» GET MORE DUSTY OLD CODE... Subscribe now at **http://bit.ly/LinuxFormat**

LXF316 will be on sale Tuesday 28th May 2024

ULTIMATE DESKTOP UPGRADE!

KDE Plasma 6 is out. We take a look at the latest and greatest desktop upgrade and what it can do for you.

Write that book!
Word processors that can help craft that novel you've always been talking about and organise large projects.

Play by mail
Revive the old roleplaying system for a digital age as we recreate our own play-by-mail gaming service!

Inside Fairphone 5
We take a look and see how ethical the latest Fairphone is and whether you should dump your Android devices.

Build the best retro console
We grab a Steam Deck (or your laptop) and Batocera, and build the ultimate retro-gaming experience.

Content of future issues subject to change. Can we interest you in Amiga Workbench 1.3?

CREDIT: Magictorch

www.linuxformat.com

The Electronic Frontier Foundation is the leading nonprofit organization defending civil liberties in the digital world. Founded in 1990, EFF champions user privacy, free expression, and innovation through impact litigation, policy analysis, grassroots activism, and technology development. We work to ensure that rights and freedoms are enhanced and protected as our use of technology grows.

EFF.ORG

ELECTRONIC FRONTIER FOUNDATION

Protecting Rights and Promoting Freedom on the Electronic Frontier

THE BRAIN TUMOUR CHARITY

A CURE CAN'T WAIT

BRAIN TUMOURS MOVE FAST. WITH YOUR HELP, WE CAN TOO!

We're working to create a future where brain tumours are curable. We urgently need your help to accelerate research.

Text DEFEAT5 to 70507 to donate £5, please help us to find a cure.

thebraintumourcharity.org

FR Registered with FUNDRAISING REGULATOR

9021

9000