MINT BACKUPS Protect your system and personal files

BLENDOS Is this the ultimate mix of Linux distros?

NEW DESKTOPS The easy way to try fresh interfaces

LINUX FORMAT

The #1 open source mag

Ubuntu spins head-to-head

CONTROL YOUR OWN A.I.

Take over your world with local open source machine learning!

65 pages of help & features

TUTORIALS

- Use HATs on your Pi Pico
- Create a Linux recording studio
- Code better terminal progress bars

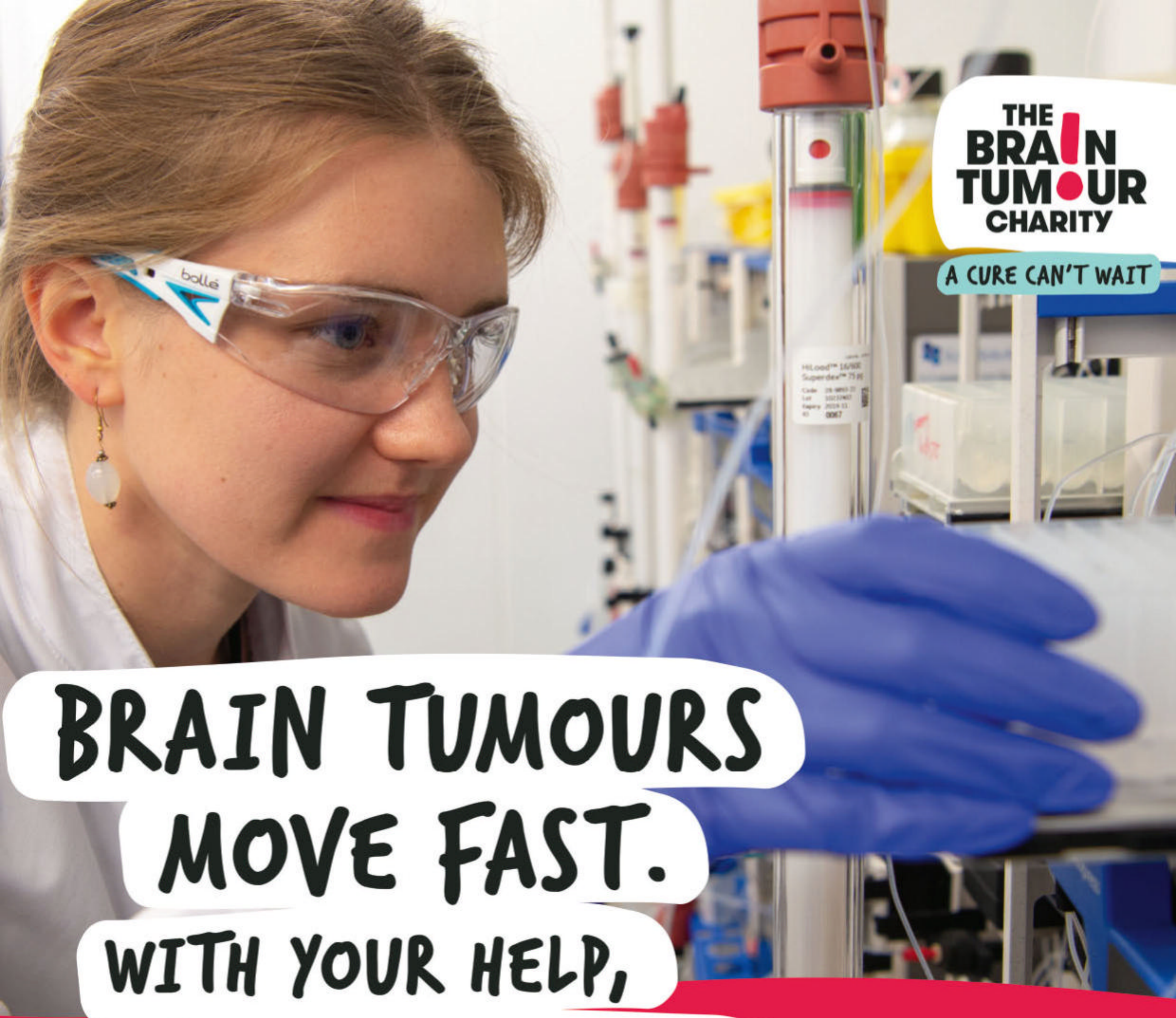
RESTORE PHOTOS GIMP and scanners help save your photos

INSIDE CPUS The new world order of processor tech

LINUX ADVENTURE Code your own old-school text adventure

LXF August 2023

FUTURE



LINUX FORMAT

» MEET THE TEAM

We're running our own home AIs this issue, so what's the one job you'd like AI to take off your hands? And it can't be dealing with grumpy editors…

Nate Drake
I'd love AI to handle my laundry. It could navigate the linen labyrinth, decode cryptic care labels, and magically fold everything. Plus, I wouldn't mind if my socks disappear – it'd probably just be having its own robot dance party, so long as it returns them to the drawer after.

Les Pounder
After a recent water leak at home, I'd love an AI home network that could monitor all the areas of my house and give me an alert before I hear the rush of water. The sensors are there, all I need to do is train the AI to be a plumber.

Mayank Sharma
Since it's tax season for me, I'd give anything for AI to prepare and file taxes. In fact, an AI that can respond to notices and do all the paperwork to satisfy the desk jockeys at government offices would be worth its weight in gold, in any part of the world.

Michael Reed
I'm looking forward to better photo search tools that use AI. It would be nice if I could search for something like 'beach sunset autumn'. In fact, all desktop search could be enhanced with tech like that. Imagine if I could search for 'bands that sound like The Cure'…

Nick Peers
The only productive use for AI I can think of is intelligent upscaling of all my standard-definition media to HD, particularly shows that will never get a remaster, such as *Star Trek: Deep Space Nine* and *Voyager*. If it could take on the washing-up, too, that would be great.

Machines learning

The Management demanded an AI cover and here we are. It's highly likely a machine told them it was a good idea and, to be fair, it is a hot topic right now.

Linux Format has looked at machine learning multiple times in the past, but the latest leap forward with ChatGTP 3 in 2022 and more recently v4 has really fired up imaginations – largely with regard to how corporations can improve profits by making swathes of humans unemployed. From dreary support chat to medical advice, legal challenges and automated news writing to code, art, music and video creation, there's currently a lot of eyes and puny human minds figuring out just how to take advantage of this new technology avenue.

Of course, this all runs on top of Linux as it's designed to be rolled out on the server farms of Google, Amazon et al, but this means that with a bit (250GB!) of downloading, you can play around with your own TUX 9000 chatbot, auto translator and robot artist at home. Let's just not think too long about the ethics of that training data…

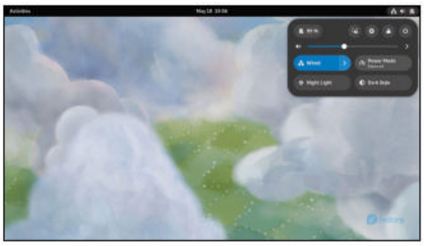
But good old human creativity is still here, so we also look at composing your own music, rescuing those old analogue photos and perhaps writing your own adventure code, plus essentials such as backing up and testing new desktops, so ensure you enjoy it while you can!

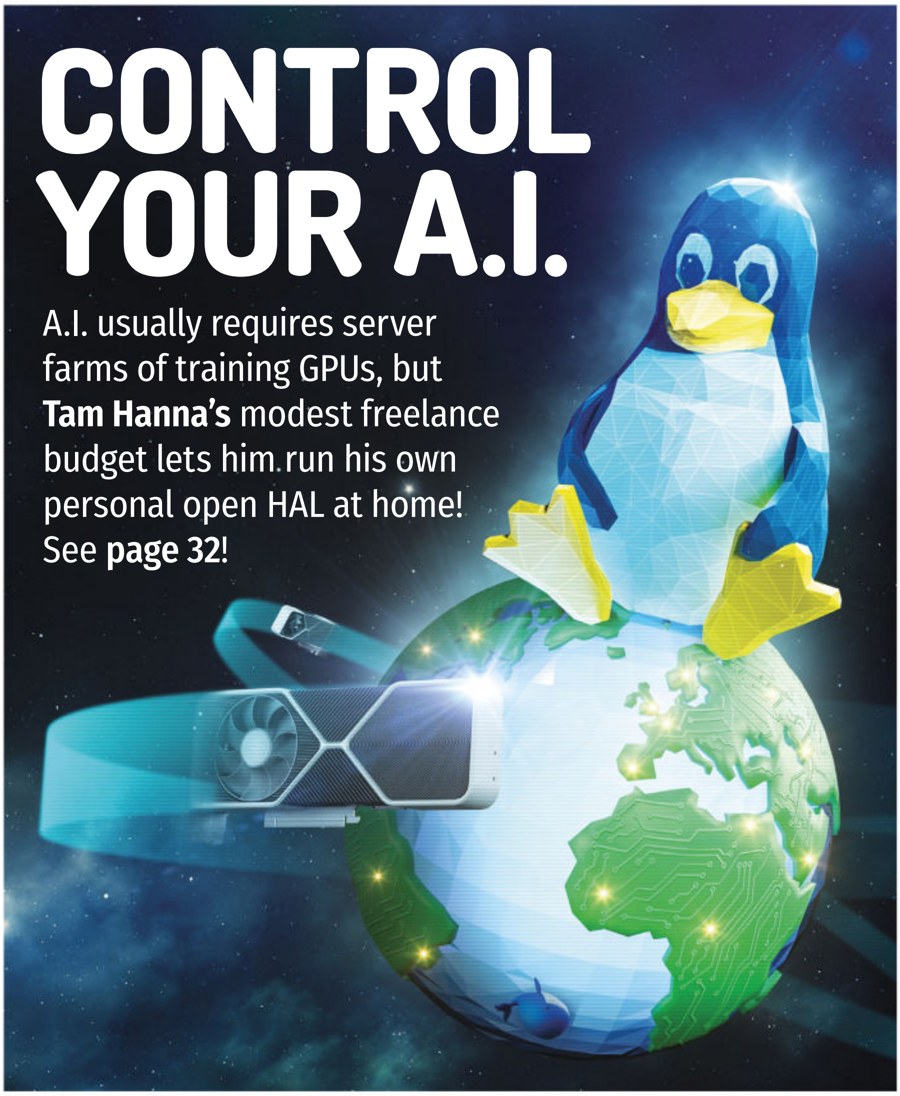
Neil

Neil Mohr Editor
neil.mohr@futurenet.com

Contents

REVIEWS



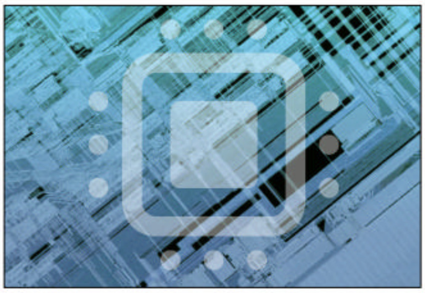


CONTROL YOUR A.I.

A.I. usually requires server farms of training GPUs, but **Tam Hanna's** modest freelance budget lets him run his own personal open HAL at home! See **page 32**!

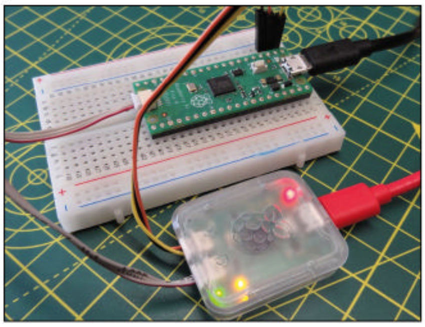
CREDIT: Magictorch

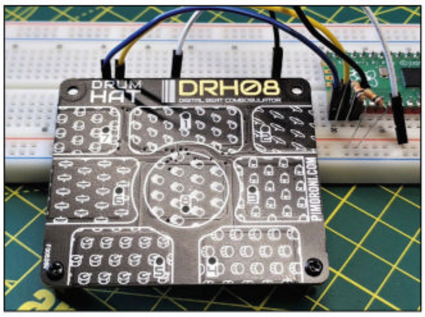
IN DEPTH



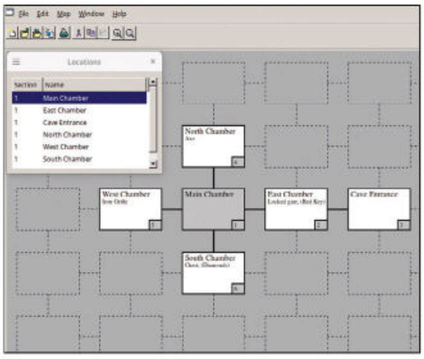
CONTENTS

Pi USER





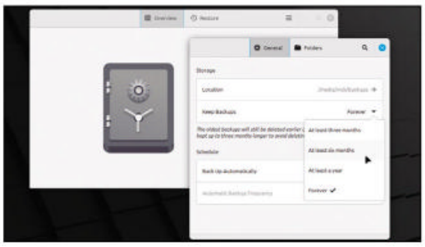
CODING ACADEMY



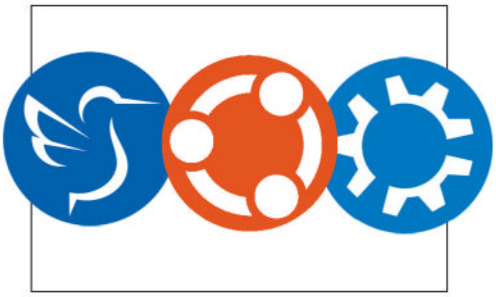
Section	Name
1	Main Chamber
1	East Chamber
1	Cave Entrance
1	North Chamber
1	West Chamber
1	South Chamber

REGULARS AT A GLANCE

TUTORIALS



ROUNDUP



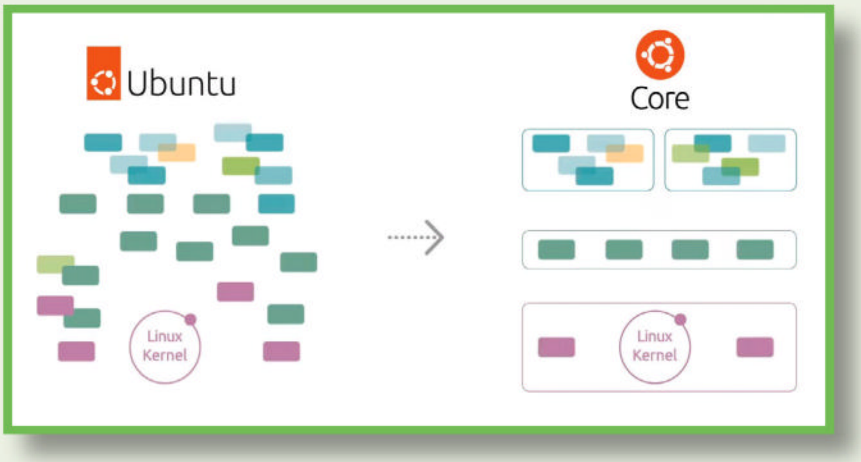
Newsdesk

THIS ISSUE: Ubunto Core is Snapped up » AMD announces free firmware » Fresh features for Gnome » Wine owned by employees

DISTRIBUTION

All Snaps – no Apt!

Canonical is cooking up a desktop version of its immutable Ubuntu Core OS with package management handled only by Snaps. Will it catch on?



Ubuntu Core is an immutable OS designed specifically for use with IoT. Applications cannot alter the underlying system code or other running programs.

On 31st May, Canonical product manager Oliver Smith raised the possibility on the Ubuntu blog of an all-Snap immutable desktop version of Ubuntu Core. Canonical even has a testing version of its popular Ubuntu Core incorporating the GDM display manager on GitHub (**https://github.com/canonical/ubuntu-core-desktop**) .

Ubuntu Core dates back to 2014 and was designed to create a fully containerised platform for IoT, using the same kernel container tech as *Docker* and *LXC.* The advantage is that every system component is sandboxed and follows a defined upgrade and rollback procedure. Updates for IoT devices are automated and the OS itself is virtually tamper-proof. Apps only see system data that they need to run. They cannot affect the core system or other applications.

This is why Ubuntu Core is known as immutable, because a running system cannot be modified directly by users or applications, and all updates are applied automatically.

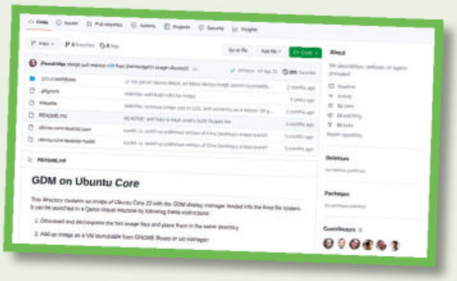
Although the security and stability benefits of an immutable OS are obvious, creating a user-friendly desktop version of Ubuntu Core could prove tricky for Canonical.

For instance, Snaps can be considered to be immutable applications. They're installed as complete, self-contained packages containing all dependencies. They can be run in a strict environment where they can't modify or access systemwide resources, and during upgrades the entire package is replaced automatically.

This would explain why an all-Snap desktop version is seemingly under development, but feedback has been lukewarm. Some users have cited the long startup time for Snaps using the default xz compression. Canonical has since switched to LZO compression, which results in faster startup times but larger packages.

Other Redditors are curious about how this desktop version of Ubuntu Core will go full Snap, asking if the OS image will be one big Snap containing different system components. This would make it harder to make changes to the base image, as you can with other immutable OSes like Fedora Silverblue.

Ultimately, whether a desktop version of the immutable Ubuntu Core OS proves popular is a matter for the Linux public at large. Despite some trepidation on discussion groups, however, there's no suggestion that the regular flavours of Ubuntu Desktop are going anywhere any time soon. The most recent release (23.04) of the OS did drop support for Flatpaks and is clearly pushing installation of packages via Snap via *Software,* but at the time of writing, you can manually re-enable Flatpaks and/or install programs via good old `apt-get` .



GDM on Ubuntu Core

This directory contains an image of Ubuntu Core 22 with the GDM display manager loaded into the boot file system. It can be launched in a Qemu virtual machine by following these instructions:

1. Download and decompress the two image files and place them in the same directory
2. Add an image as a VM launchable from GNOME Boxes or virt-manager

The testing version of Ubuntu Core with GDM on GitHub, along with instructions to test it in a Qemu VM.

CREDIT: Ubuntu, GitHub, AMD

FIRMWARE

AMD to embrace open system firmware?

Tech firm has revealed the release of OpenSIL.

AMD has announced plans to release OpenSIL (Open-source Silicon Initialization Library), complete with AMD Coreboot support. In theory, this will free users and manufacturers from the shackles of proprietary UEFI/BIOS firmware, allowing better detection of bugs and better compatibility. Open source enthusiasts also may appreciate the assurance that truly free firmware reduces the attack surface for bad actors to conceal hardware backdoors to allow spying.

If this sounds familiar, it's because in 2011, AMD promised to support the open source Linux-based BIOS implementation Coreboot on all future processors. However, support wasn't universal across all AMD CPUs and mobos.

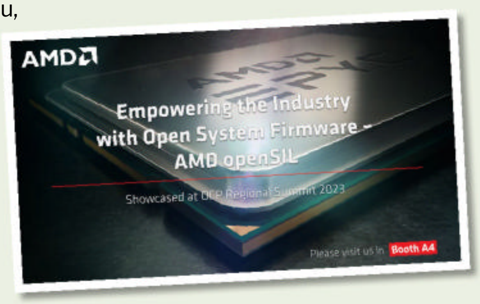
In 2014 AMD also released AMDgpu, an open source Linux driver for its Radeon lineup of graphics cards.

In the same blog post, AMD chief firmware architect Raj Kapoor elucidated: "AMD OpenSIL is a set of three statically linked libraries – xSIM (x86 Silicon Initialization Libraries), xPRF (x86 Platform Reference Library) and xUSL (x86 Utilities and Services Library) that can be statically linked to any host firmware during compile/link time."

The host firmware in question apparently can include open source solutions such as Coreboot, Oreboot, FortiBIOS and Project Mu.

Currently, OpenSIL is only available as proof of concept. The OpenSIL Firmware Architecture Specification has yet to be published in full.

Response from the coding community has generally been positive, including comparisons with the 2022 implementation of AMD's own adoption of its Genoa processors for use with reference servers in conjunction with OpenBMC, an open source distro whose goal is to produce the implementation of baseboard management controllers without proprietary code.


AMD
Empowering the Industry with Open System Firmware - AMD openSIL
Showcased at OCP Regional Summit 2023
Please visit us in Booth A4

OpenSIL was announced at the Open Compute Project Summit 2023. Time will tell if it means full Coreboot compatibility.

DESKTOP ENVIRONMENT

Gnome Kuala Lumpur updated

Gnome 44.2 fixes bugs and adds new features.

A second update (44.2) has been released for the Gnome 44 Kuala Lumpur desktop environment series. The Gnome Project has encouraged anyone running Gnome 44 to upgrade.

The latest version includes checking for updates after login if the first check fails for any reason. It also reduces background resources used by Gnome whilst idle.

The *Nautilus* file manager has also been updated and includes fixes for a number of bugs, including better support for displaying expandable folders. The latest update also includes *Gnome Maps 44.2*. This includes tweaks such as showing the correct rotation of the user location marker when there is a heading, as well as displaying thumbnails for places with a Wikidata entry with a title image.

The Touchpad row can also now be hidden in cases where the Touchpad can't be disabled, and a minor bug with racy radio buttons in the *Network* connection editor has been fixed.

The *Epiphany* web browser has been upgraded to version 44.3 and addresses several bugs, while the latest version of *Gnome Boxes* (44.2) now enables users to preserve run-in-background and 3D acceleration configurations for clones of any virtual machines they create. It also fixes an issue where shared folders created for VMs sometimes didn't appear.

View the release notes at: **https://discourse.gnome.org/t/gnome-44-2-released/15563**

OPINION

JOIN THE GANG!

Mike Saunders wrote for **LXF1** (and many more issues) and now works for The Document Foundation.

"Large, well-established open source projects can be daunting for potential new contributors. What's the best way to get involved? Why are there so many different channels of communication? And who should you contact, if you need help?

This is a challenge we face in the LibreOffice community. The codebase is seven million lines of predominantly C++, with a bunch of dependencies. But we've been grafting to simplify the build process, and have taken on development mentors to help new coders dip their toes in the pond (or ocean).

There are many other ways to get involved, including UI design, marketing, quality assurance, documentation, translations and infrastructure – and in May, we organised a campaign to convert users into contributors. The Month of LibreOffice rewarded everyone who helped out with a sticker pack.

During the month, 338 people joined in, reflecting a healthy *LibreOffice* community. (And that is just volunteers, for the most part – there are many others on the commercial side.)

See **https://tinyurl.com/mol-may2023** for the results and stay tuned for the next Month of LibreOffice, in November!"

OPINION

HEAD WESTON!

Marius Vlad
is a senior software engineer at Collabora.

“Weston is a Wayland compositor. Released in mid-May, v12.0 brought multiple enhancements including two new back-ends, support for multiple scanout devices and the addition of new protocol implementations. Alongside these features, we also have been adding multiple fixes and internal changes that would further facilitate integration of functionality like colour management or the ability to load up multiple back-ends at the same time.

The backend-drm has seen a lot of changes: plane rotation, alpha and HDMI content type properties have been integrated. Depending on the client’s content buffer, the compositor can leverage properties to make use of available display hardware capabilities. Also, support for multiple GPUs as scanout devices (KMS) has landed and is enabled by passing additional devices when starting up the compositor.

As we head towards the ability to load multiple back-ends, two new ones have seen the light of day. Backend-vnc, which is similar to backend-rdp, is based on aml and neatvnc libraries. It has TLS support and user authentication, while the other is a PipeWire back-end that creates a node for each output and can be used to capture Weston outputs for processing with other applications.”

SOFTWARE

Wine to be employee-owned

CodeWeavers establishes a trust.

The Proton compatibility layer for Steam Deck is based on CodeWeavers’ Wine project.

CodeWeavers is establishing an employee ownership trust that will hold a controlling stake in the software company.

This is exciting news as the company is behind the open source *Wine* project, which allows many Windows programs to run on Linux. It also powers the *Proton* compatibility layer used by the Steam Deck.

Under the new ownership structure, the trust will eventually become the sole shareholder of the company. In the meantime, former CodeWeavers CEO Jeremy White has stood down and now serves as chairman of the board and a member of the Trust Stewardship Committee. He will remain a shareholder for seven years, after which the trust will be the sole owner.

In a CodeWeavers press release, White explained: “The trust allows the company to be run for the benefit of the staff and the broader community, rather than for shareholder profit. I believe this will create a more sustainable, ethical and equitable business.”

The same release reiterated CodeWeavers’ dedication to open source principles.

DISTRO

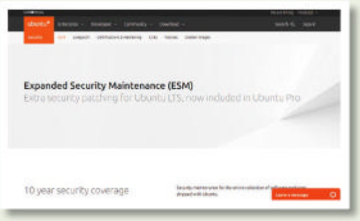
Ubuntu 18.04 moves to ESM

Bionic Beaver gets a new lease of life thanks to premium features.

Most Ubuntu users are familiar with the LTS (Long Term Support) releases, which are officially supported by Canonical for five years after release.

Officially, Ubuntu 18.04 (Bionic Beaver) reached its end of life this April. However, Canonical is offering Extended Security Maintenance (ESM) for the OS, just as it did for Ubuntu 16.04 LTS.

This means businesses who haven’t yet made the upgrade to the latest LTS release can still receive critical security updates. Canonical recommends the best way to do this is through an Ubuntu Pro subscription.



Expanded Security Maintenance (ESM)

Extra security patching for Ubuntu LTS, now included in Ubuntu Pro

10 year security coverage

ESM via Ubuntu Pro provides 10-year coverage for all Ubuntu software packages.

SOFTWARE

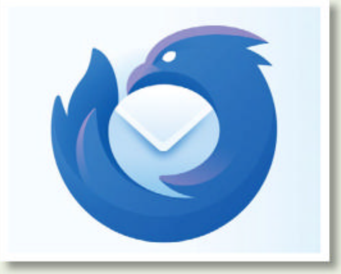
Thunderbird gets new logo

5, 4, 3, 2, 1... Thunderbird email client has an overhaul and a new logo.

In late May, Ryan Sipes, a product manager at Thunderbird, announced that the famous email client is “enjoying a resurgence”.

As part of a revamp, 20 years after designing the first *Thunderbird* logo, Jon Hicks was asked to reimagine what Sipes calls his original “wig on an envelope” design.

Initial feedback has been positive, with users comparing it to the revamped *Firefox* logo. Read more about latest *Thunderbird* developments at **https://blog.thunderbird.net/**.



The icon pictured here will be used for both the Windows and Linux versions of Thunderbird.

CREDIT wikipedia/commons, Thunderbird

Distro watch

What's behind the free software sofa?

ARMBIAN 23.05.1

One of the few Linux projects to come out of Slovenia, Armbian has been around for about 10 years but in that time there have only been two stable releases. The focus of Armbian is on producing a Linux distribution for ARM development boards, such as the Banana Pi, Orange Pi, Pine 64 and others. There's also a generic AArch64 image.

The latest version of Armbian, 23.05 (code name Suni), has an LTS kernel based around Debian 12 and Ubuntu 22.04.

Designed for ARM development boards, Armbian supports a wide range of devices.

MX LINUX 23 BETA 1

Following two unannounced alpha releases, the first beta of MX Linux 23 has been officially released. The OS is always based on the latest Debian stable branch (currently Debian 12 Bookworm) and MX repositories. By default, it uses the Xfce desktop environment (currently v4.18) but there are also spins available using Fluxbox 1.3.7 and KDE/Plasma 5.27.

The latest version's live boot menu now enables users to check media. The installer also supports both swap files and swap partitions.

MX is a mid-weight OS based on the latest stable branch of Debian.

ARCOLINUX 23.05.4

ArcoLinux is based on Arch Linux and uses the *Calamares* installer. Like Arch, it follows a rolling release model. The main release, ArcoLinuxL, is fully-featured, shipping with the Xfce desktop, *OpenBox* and *i3* window managers, and a complete software suite.

There's also a selection of minimalist ISOs, such as ArcoLinuxD, which comes with no desktop environment or preinstalled software, so you can tailor the operating system to your needs. The main website has extensive video tutorials on how to do this.

Arco can be downloaded and installed graphically, or you can customise your ISO.

ATHENA OS 2023.06.01 PARTHENOS

Athena is also based on Arch Linux. However, its main focus is on helping users to learn how to master cybersecurity.

This Swiss-based Linux newcomer has access to the infamous BlackArch repository of pen-testing tools but describes itself as "lightweight", in that the bare minimum of software is preinstalled.

The latest version (code name Parthenos) integrates Hack the Box VIP content for playing retired machines. It also comes with an OpenAI ChatGPT desktop client. The default desktop environment is Gnome.

Arm yourself with Athena and learn vital skills for cybersecurity proficiency.

OPINION

SPREAD THE WORD

Joe Brockmeier is head of community at Percona.

"WordPress turned 20 recently, which is a good opportunity to take stock of the impact it's had.

It has evolved from a nifty, easy-to-use blogging platform for enthusiasts to a ubiquitous tool that powers a huge chunk of the web.

Along the way, WordPress has been a fantastic ambassador for open source. People who wouldn't have any reason to touch desktop Linux or do programming still wanted to run a blog or website. So they got hands-on with WordPress and, generally, had a good experience. Even better, WordPress is a really friendly, helpful community.

It's also had a symbiotic relationship with *MySQL*. Millions of people who'd never have any other reason to work with a database got their feet wet standing up a *MySQL* database for WordPress. And *MySQL*, part of the LAMP stack that powers WordPress, helped WordPress spread.

MySQL is powerful enough to run sites with massive amounts of traffic but is also resource friendly and easy to use for millions of users with more modest needs.

For more than 20 years, WordPress, with *MySQL*, has given millions and millions of people a voice and a home on the internet. It's an amazing achievement and I can't wait to see what the next 20 years bring!"

OPINION

WINDOWS -WISE

Jon Masters has been involved with Linux for over 22 years.

"I recently hosted an industry gathering focused on server standards. I got chatting with a friend from a major chip company who has been responsible over the years for helping to lead 'bring-up' of Windows and Linux on new chips. When a chip comes back from the factory, the first few weeks are when the logic blocks are tested, cores, fabrics, IOs and so on are gotten working, and everything is in some kind of organised chaos with people running round in labs.

My friend told me something interesting: he said that Windows was a true test for the engineers, because it is not open source. When it comes to Linux, you can consult the source, or ping a friendly kernel dev, so the way he ended up truly learning every last piece of the machine came when he had to disassemble Windows code and figure out what it was trying to do from first principles, looking at disassembled binaries.

The kind you get from running *Objdump* on an (ELF) program, or via a tool like the NSA's (yes, that NSA) Ghidra reverse engineering framework, used in combination with the compiler explorer **godbolt.org** (after Matt Godbolt) showing how different compilers compile code. Consider walking through a kernel binary – it's revealing!"

Kernel Watch

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

Linus Torvalds announced Linux 6.4-rc6, saying, "I don't think we've had anything hugely interesting happen the last week, and the whole 6.4 release really does feel like it's going fairly smoothly. Knock wood, famous last words, you know the drill." This contrasts with the previous rc5 release, in which Linus had to revert a well-intended fix to module loading that turned out to break some systems. If things continue on the current track, we'll likely be covering the 6.5 merge window (period of time during which disruptive patches are merged for a new kernel release cycle) in next month's edition.

Finally this month, I would like to give a shout-out to the coverage over at *Linux Weekly News* of the recent Linux Storage, Filesystem, Memory-Management and BPF Summit. Their coverage includes several read-outs on ongoing developments with CXL (Compute Express Link), in some respects a successor to PCI Express that allows for such innovations as memory expansion via CXL. Among the use cases covered was live migration of VMs without copying memory by leveraging shared CXL memory pools at the data centre level.

These days, Linux is increasingly used in real-time environments, such as those involving industrial automation (Linus once joked about crazy people using Linux to control laser-welding robots) or stock market high-frequency trading. Indeed, a multi-decade effort has been underway to upstream the PREEMPT-RT patches, with many pieces of these finally getting into Linus's tree over the past few years. One of the key issues such patches aim to solve is that of priority inversion, in which a low-priority task (such as a simple bookkeeping task) can block a real-time task from making progress when needed.

This is actually what happened in the NASA Mars Pathfinder programme (though it was running a commercial RTOS and not Linux). Typically, priority inversion is solved by temporarily 'boosting' (inheriting) the priority of a dependent but otherwise low-priority task to that of the real-time task depending upon its output. Thus, when a critical lock is needed by a real-time task, the non-RT task might be temporarily boosted so that it can release the required lock.

As John Stultz put it in his email about proxy execution, this "is a generalised form of priority inheritance". It's interesting to watch as it might really improve the overall capability of Linux to handle real time. This won't just benefit NASA rovers, but also audio engineers and gaming fanatics running Linux. LXF

» ONGOING DEVELOPMENT...

Randy Dunlap reminded Linus (and everyone else) of the existence of the **scripts/parse-maintainers.pl** script for sorting the entries in the kernel **MAINTAINERS** file. He noted that "it was giant, more than 100 out-of-order entries since it was last sorted". Linus responded, "Ugh. That file causes the most conflicts, and sorting it makes it horrendous." A suggestion to sort each RC1 was made (because Linus effectively owns the tree at that point).

Matthew 'Willy' Wilcox posted "Futexes & Folios" with an initial patch showing how he might convert the kernel's futex locking code to use his newer folios rather than the legacy page structures. Hardware MMUs (Memory Management Units) as found in CPUs, for example, (and GPUs and so on) operate upon units of memory sized in pages, typically these are 4096 bytes (4K). Folios aim to arbitrarily group these pages together into larger chunks that are easier to manage, where possible. Futexes (fast mutexes) are commonly used to back pthread mutexes used by Linux application code. Migrating the kernel over to folios is an ongoing journey.

More patches continue to trickle in for the StarFive RISC-V-based SoCs (system-on-chips) commonly used in Pi-like boards for that architecture. This month's patches were for the camera subsystem used on those SoCs. The upstream story appears to be fairly good at this point for these low-cost developer boards.

Answers

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

Neil Bothwick
lives in a Tux shaped house powered by Linux, honest!

Q Virtually full disk

I have a fairly old version of Linux Mint Xfce installed in a *VirtualBox* virtual machine. The size of the virtual disk is roughly 20GB and now 18GB has been taken up. How can I find big packages so I can see whether they have been used, and remove them as needed?
Cameron Wright

A Are you running out of space on the virtual disk or on the physical disk holding it? *VirtualBox* creates dynamically sized virtual disks by default. This means that if you create a 20GB disk and load 10GB of data on to it, the virtual disk file is only about 10GB in size. As you add to it, the file grows, but bear in mind that if you remove data from the virtual disk, the file does not shrink. So, a size of 18GB only means it has held that much data at some time. If you delete, say, 5GB, the file is still 18GB even though it only holds 13GB. That space is not lost, though, as it is used as you add further data. So, the first step is to check how much is actually in use by running `$ df -h /` in a terminal within the virtual machine.

If it is significantly less than the 18GB taken up by the disk file, you can resize the virtual disk file to reclaim the unused space, but it will subsequently grow as you write to the virtual disk. On the opposite side of the equation, if the virtual disk is nearly full but you have plenty of space on the host system, you can enlarge the virtual disk in the Virtual Media Manager section of *VirtualBox*. Do this while the VM is not running. When you start it up, the extra space is not available to you until you open the partition manager or *GParted* and resize the root partition.

If you need to reclaim space within the virtual machine, you should first look at cache directories, especially **.cache** in your **home** directory and **/var/cache/apt/archives**. The latter contains package files downloaded by the package manager. These can be safely deleted. Once you have cleared out old files, turn your attention to installed packages and get a size-ordered listing using *Aptitude*. This may not be installed, but you can install it from the package manager and it does not need much space. It is a standard tool for package manipulation on Debian-based systems. Then open a terminal and run this command:

```
$ aptitude search --sort '~installsize'
--display-format '%I %p' '~i' | less
```

The `--sort` option tells it to sort the output on installed size, the `~` switches the sort to descending size from the default of ascending (you want the space hogs first). The `--display-format` option sets the output to the package size followed by its name, and the `~i` flag restricts the search to installed packages. We pipe the output through `less` to read the output in a pager – you could also redirect it to a file. The quotes are there to stop the shell trying to interpret the `~` as part of a user specification.

Once you have this list you can decide which packages you can do without and uninstall them, but some will not uninstall if they are needed by other packages.

If you want to shrink the virtual disk file, do so after removing everything you don't need. First you must fill the unused space with zeros, so open a terminal and run:

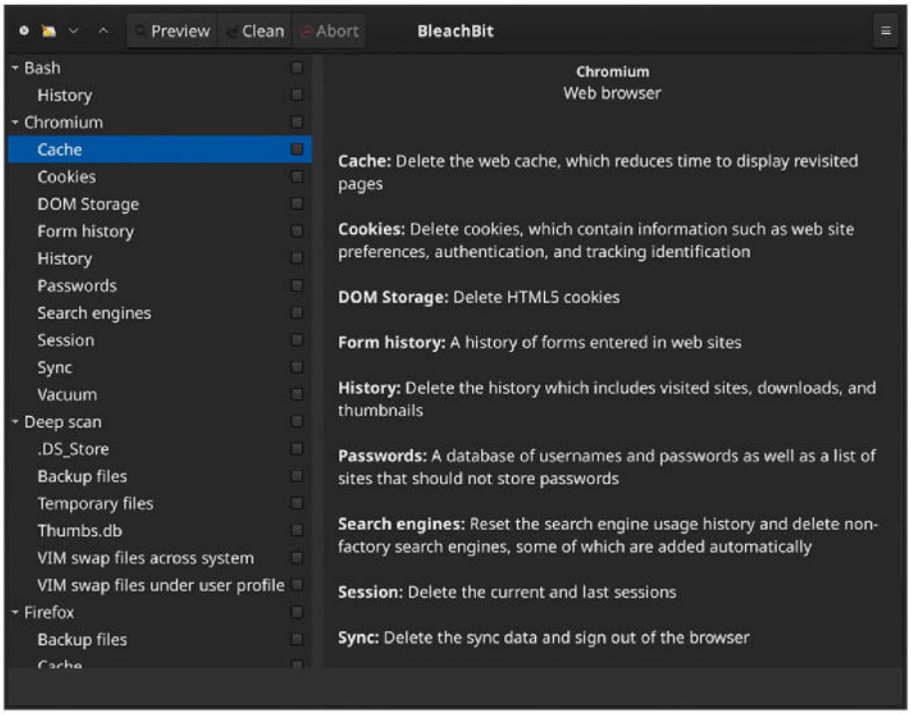
```
$ dd if=/dev/zero of=zerofile
```

You can do this as a normal user in your **home** directory. It exits with an error once there is no more free space, which you can check with `df`. Delete **zerofile** – Linux hates running on a full disk – and power down. Open a terminal on the host and run:

```
$ vboxmanage list hdds
```

Identify the virtual disk you wish to shrink and run:

```
$ vboxmanage modifymedium disk /
path/to/virtual/disk --compact
```



Chromium
Web browser

Cache: Delete the web cache, which reduces time to display revisited pages

Cookies: Delete cookies, which contain information such as web site preferences, authentication, and tracking identification

DOM Storage: Delete HTML5 cookies

Form history: A history of forms entered in web sites

History: Delete the history which includes visited sites, downloads, and thumbnails

Passwords: A database of usernames and passwords as well as a list of sites that should not store passwords

Search engines: Reset the search engine usage history and delete non-factory search engines, some of which are added automatically

Session: Delete the current and last sessions

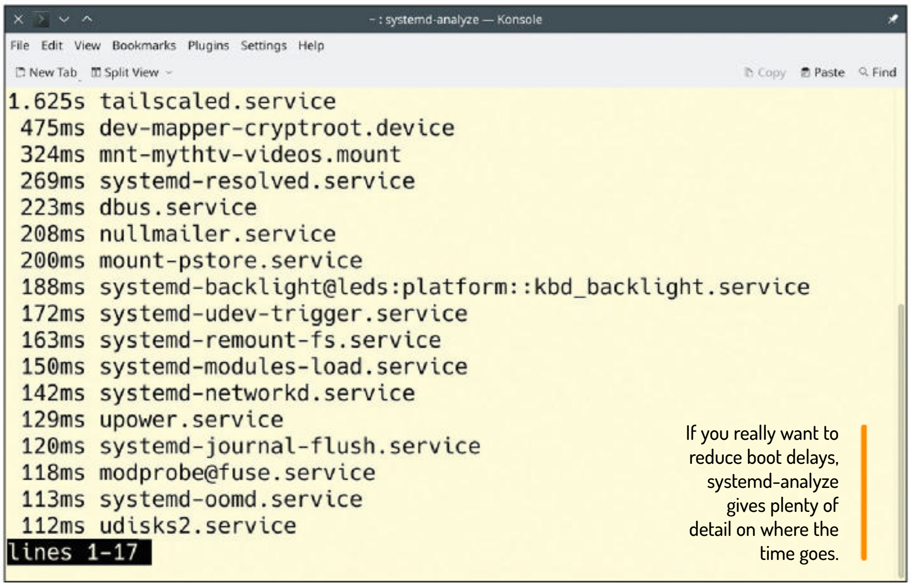
Sync: Delete the sync data and sign out of the browser

BleachBit (www.bleachbit.org) is a good way to identify and remove cruft from your hard drive.

Q Cryptic sizing

I have upgraded the M.2 drive in my Debian laptop from 500GB to 1TB using *CloneZilla*. Now I have 500GB unallocated and would like to extend my main LUKS partition to the remaining space. Searching around, I found rather confusing info. Here is the current layout:

```
nvme0n1p1   300M  /boot/efi
nvme0n1p2   460G  encrypted root
```


~ : systemd-analyze — Konsole
File Edit View Bookmarks Plugins Settings Help
New Tab
Split View
Copy
Paste
Find

```
1.625s tailscaled.service
 475ms dev-mapper-cryptroot.device
 324ms mnt-mythtv-videos.mount
 269ms systemd-resolved.service
 223ms dbus.service
 208ms nullmailer.service
 200ms mount-pstore.service
 188ms systemd-backlight@leds:platform::kbd_backlight.service
 172ms systemd-udev-trigger.service
 163ms systemd-remount-fs.service
 150ms systemd-modules-load.service
 142ms systemd-networkd.service
 129ms upower.service
 120ms systemd-journal-flush.service
 118ms modprobe@fuse.service
 113ms systemd-oomd.service
 112ms udisks2.service
lines 1-17
```

If you really want to reduce boot delays, systemd-analyze gives plenty of detail on where the time goes.

```
nvme0n1p3   16G  encrypted swap
unallocated  500G
```

Mason Palmer

A This is a simple task, but it is complicated by your **swap** partition that is now in the middle of the drive. Removing the swap area before resizing the **root** partition, then recreating it after is the simplest solution. Providing you do not do anything memory-intensive while resizing, this should not be a problem. If you are concerned about this, you could create a swap file for the duration of the process. Alternatively, you could do everything from a live distro; enlarging filesystems is generally safe to do on a live system. Bear in mind that any procedure involving filesystems carries a small risk of corruption, but everything important should already be backed up. All of these actions should be performed as root or prefixed with `sudo`. First get rid of **swap**:

```
$ swapon -s
$ swapoff /dev/mapper/CRYPTSWAP
$ cryptsetup close CRYPTSWAP
```

`CRYPTSWAP` is the name of the encrypted swap container, which should be returned by the first command. Now use your preferred partitioning tool to delete the **swap** partition and create a new one at the end of the disk. Then delete and recreate the **root** partition. Don't worry that you are currently using this partition, tools like *gdisk* and *parted* do not write the updated partition table to the disk until you tell them to. Now you have a new partition starting in exactly the same place as the original, just bigger.

The next step is to resize the encrypted container to fill the new partition, then resize the filesystem to fill the container. Both commands are given with no size as they default to using all available space:

```
$ cryptsetup resize CRYPTROOT
$ resize2fs /dev/mapper/CRYPTROOT
```

Replace `CRYPTROOT` with the name of your encrypted root container. If you are using a filesystem other than ext4, consult its documentation for the command to resize it. The `df` command should confirm that your **root** filesystem is now the size you expect. Lastly, create a new container for **swap** and initialise it:

```
$ cryptsetup luksFormat /dev/
nvme0n1p3
$ cryptsetup luksOpen /dev/nvme0n1p3
CRYPTROOT
$ mkswap /dev/mapper/CRYPTROOT
$ swapon /dev/mapper/CRYPTROOT
```

Finish off by updating the entry in **/etc/fstab** to match the name you just gave it, unless you reused the old name.

A QUICK REFERENCE TO... TREE

Graphical file managers let you view the contents of a directory or one of its subdirectories, as do the `ls` and `dir` commands in the shell. Most graphical managers also let you view directories in a tree format, where you can see both subdirectories and their contents in a single view. There is an often-overlooked shell command that does much the same and it goes by the unsurprising name of `tree`.

Run with no arguments, `tree` prints a listing of the files and directories in the current directory and those below it. As with `ls`, you can pass the name of a directory to list and there are several options that alter the output. Some of the most useful are in the table.

There are plenty more options, including the ability to output in XML, HTML or JSON, control the colourisation of output and much more. The man page details them all. It's not a replacement for `ls`, but `tree` is a more of a complement to it, each of the commands performing useful but different functions.

Option	Function
a	Shows hidden dotfiles
L N	Only goes N directories deep – useful for something like a home directory with lots of nested subdirectories
D	Shows the last modification time of each file
p	Shows each file's type and permissions
h	Shows the size of each file in human-readable format
prune	Hides empty directories
P <pattern>	Shows only files that match the given wildcard pattern
f	Prints the full path for each file

Movable disks

Q I have several Linux distros on my machine. Often when I hook up an extra hard drive or something, the drive order changes, and some *Systemd* service can't find a filesystem, because it was sdc15 and now it's something else. Then I have to wait 90 seconds for it to give up. I've grepped all through /etc and can't find any reference to that timeout, and I want to change it to five seconds or so. I don't speak *Systemd*, but I have 10 or 15 Linux OSes that use it that I boot occasionally.

George Bates

A You can change this timeout in **/etc/systemd/system.conf** – the setting is **DefaultDeviceTimeoutSec**. You can also change it for individual devices in **/etc/fstab** by setting **x-systemd.device-timeout** in the Options field. However, neither of these solves your problem, they just make the symptom of it 85 seconds less annoying. Also, you'd need to make the change on each distro.

You must specify the devices to mount using factors that don't change. The two most useful are UUID and LABEL. Each filesystem has a UUID (universally unique identifier, although it's not strictly unique) that you can see with `lsblk`:

```
$ lsblk -o UUID /dev/sdc15
```

Now you can use the UUID instead of the device name in **fstab**, like this:

```
UUID=xxxxxx /mountpoint....
```

This works, but it is not very readable, which is why distros that use UUIDs often have copious comments in **/etc/fstab**. You can solve this by using filesystem labels instead. Filesystems are not given labels when formatted unless one is specified, but you can add one later without affecting the filesystem. This is handy if running multiple distros as you can name each distro's **root** partition accordingly and know what everything is. To add a label to an ext4 filesystem, use `e2label`:

```
$ sudo e2label /dev/sdc15 ubuntu
```

If you do not specify a label, `e2label` returns the current label. Then you can mount it from **fstab** with:

```
LABEL=ubuntu /mountpoint ...
```

Q Su-per GUI

I have a shell script I need to run with root privileges. When I run it from a terminal with *sudo*, it works, but I want to be able to run it from the GUI, using an icon on a toolbar. In the past I would have used *gksu* to gain root privileges, but that is no longer present in Linux Mint. Is there an alternative that doesn't require a terminal in which to input the password?

Alex Barnett

A Yes, *gksu* is no longer supported, (although KDE users still have *kdesu*). The program to use now is *pkexec*, part of *polkit*. You can run it like this:

```
$ pkexec env DISPLAY=$DISPLAY
XAUTHORITY=$XAUTHORITY
HOME=$HOME yourscript
```

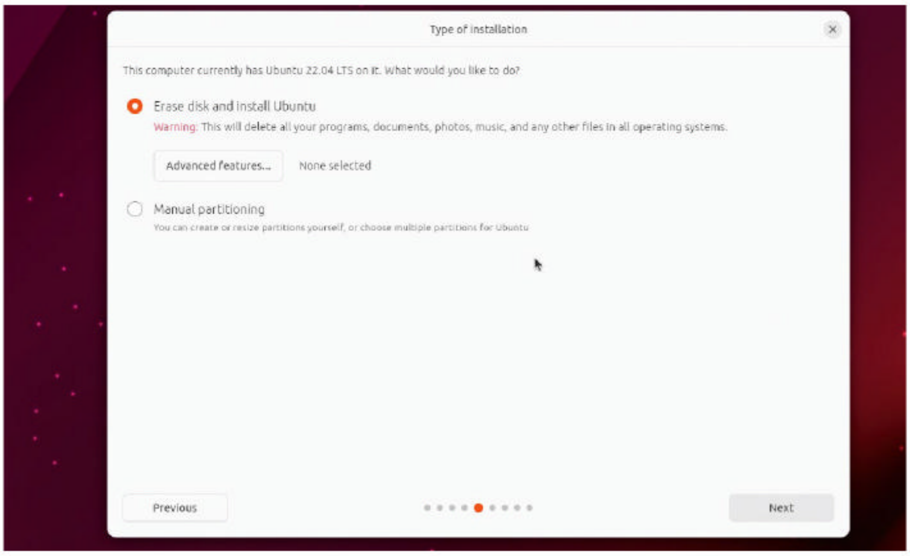
This pulls relevant settings from the environment. Similar steps are required with *sudo* unless you use `-E`, then run the program as the user specified with `--user`, or as root if no user is specified, like here.

There is another way to address this, by configuring *sudo* to allow this script to be run as root without giving a password. Providing it is acceptable to allow this script to be run as root without a password, edit the **/etc/sudoers** file, by running `$ sudo visudo`, and add this line:

```
username All=(All) NOPASSWD: /full/
path/to/your/script
```

This allows you to run your script with *sudo* without giving a password. You may need to use `sudo -E` to preserve environment variables when doing so. The script name is hard-coded into **sudoers**, which can only be edited by root, but you need to make sure the same is true of the script itself. Otherwise, someone could edit the script to do anything they want, with full root privileges. Either put it somewhere only root can edit it, such as **/usr/local/bin**, and owned by root, or make the script immutable with:

```
$ chattr +i /full/path/to/your/script
```



Type of installation

This computer currently has Ubuntu 22.04 LTS on it. What would you like to do?

● Erase disk and install Ubuntu

Warning: This will delete all your programs, documents, photos, music, and any other files in all operating systems.

Advanced features... None selected

○ Manual partitioning

You can create or resize partitions yourself, or choose multiple partitions for Ubuntu

If you want a fresh start, Ubuntu's installer, along with most others, lets you erase everything on the disk.

Q Mystery Ubuntu

I bought an old but Linux-compatible system. It's good enough for now, but the guy I bought it from had built it and loaded Ubuntu 22.04. It runs well, but I can't install any new programs or tweak the existing stuff. I can't get past authentication and can't update anything. When I picked it up, my friend said he could not remember if there was a password on it, but didn't think so. Apparently, it does and he can't remember the password.

Ava Bell

A You don't know exactly what is on this computer and, apparently, the person who installed it doesn't know either. While it is possible to reset the password on a system like this, a safer option would be a fresh installation. If you have used it, back up your **home** directory to an external drive or USB stick. There won't be anything else customised on the system as you don't have the password. Your best option is to download your preferred Ubuntu flavour, copy it to a USB stick, or a DVD if the computer has a DVD drive, and boot from it. Run the installer and select the option to use the whole disk. This erases anything already on there and sets up a fresh Ubuntu install. It asks you to set a password, the Ubuntu installer always does, when creating your user, but this is one you choose and can, hopefully, remember. All trace of the previous installation is erased. If you backed up files from your **home** directory, you can restore them after booting the new system.

If you want to reset the password without reinstalling, you need the **root** partition's username and location. Get the latter by running `$ findmnt /` in a terminal. Boot the live distro and open a terminal. Replacing `/dev/sdXN` with the root partition, mount the root filesystem with:

```
$ mkdir -p /mnt/root
$ mount /dev/sdXN /mnt/root
```

Now you can set the password with:

```
$ passwd --root /mnt/root username
```

You are prompted for a password and changes are written to the system specified by the `--root` option rather than the running system. Reboot and your user should have a brand new password. LXF

GET HELP NOW!

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

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

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

Mailserver

WRITE TO US

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

Suggestion box

You should have a column for what Linux users would like to see added to Linux distros and programs. It could feature a user's wish (with some details) and expert comment or analysis of the wish (why it can or can't be). You can perhaps give out prizes for the selected submissions of maybe a subscription or some trinkets, such as a *Linux Format* mug, pens and so on.
T Smith

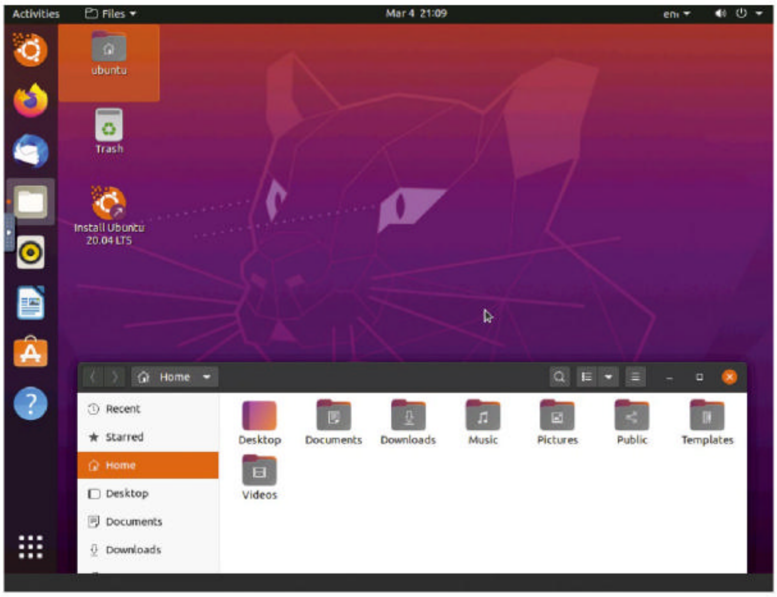
Neil says...

I'd quite like a *Linux Format* mug... It's not a bad idea, but I'm not sure the amount of correspondence we get would justify it. If you've got suggestions, we'd be more than happy to print them in *Mailserver* – after all, these pages are for readers' opinions.

Off the map

I've looked back through the archives that mention *Blender* and can find no tutorial that addresses the use of *Blender* in creating 3D images from freely available global maps (such as Google Earth and Google Maps) and freely available digital terrain model data sets that provide elevation data.

I just wondered whether Michael Reed might like to do a tutorial on this aspect of *Blender* – useful for walkers and local environmental concerns (such as flooding investigation), but also the basis for many artistic possibilities.
Colin



LXFbuntu will not be a thing, but let us know what you think should be!

Neil says...

Thanks for the suggestion. *Blender* feels a little heavyweight for this type of activity, but as long as you could transform the mapping data into something *Blender* could understand, there's no reason why it wouldn't work. I imagine there are dedicated viewing tools for this, but they likely wouldn't offer *Blender's* rendering capabilities. Michael is going to review 2D/3D CAD packages, which again might be an option, too. We did find a Google Maps importer project at **https://github.com/eliemichel/MapsModelsImporter**.

Map it

I want to create some maps with personal information overlaid, such as routes, memories and so on, and I want the overlays to be stored locally and accessible only on my local network. I use OpenSuSE and KDE. What data sources and tools would you recommend?
Chris Tucker

Neil says...

The KDE mapping tool is called *Marble Maps.* This taps into OpenStreetMaps, among many other open source tools, such as *Digikam.* That's going to be your best bet, as it provides complete options to create custom maps, even of other planets or moons, for your own use. Take a look at the documentation and see if it's any use: **https://techbase.kde.org/Marble/CustomMaps/**. There could be something more straightforward than this, though. Can anyone help?

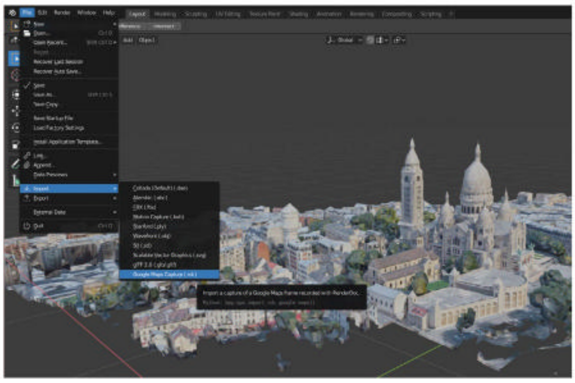
Good Outlook

I have enjoyed and learned a lot from *Linux Format* magazine for many years. I really enjoyed the **LXF297** Escape Windows article. The only thing that is holding me back from totally escaping Windows OS is that I use *Microsoft Outlook Notes.* I have over 350 notes across 15 categories that I use in *Outlook Notes.* I have already replaced *Microsoft Office* with *LibreOffice.*

If you could recommend some suitable Linux software to replace *Outlook Notes* as well as software that can convert the *Outlook Notes* extension (*.pst)

Helpdex

JOHN ! LONG TIME, NO SEE
HIYA
I HEARD THAT A YEAR AGO YOU QUIT CODING AND BECAME A FARMER ?
YES
LONG TIME, NO C
!
Shane



Creating maps in Blender is a thing, but we don't know how to do it yet...

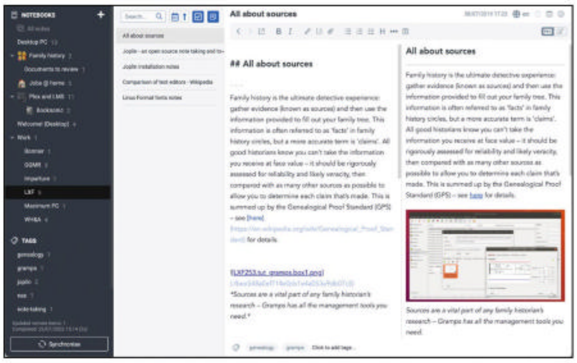
to the Linux extension, it would help me to completely transition into the Linux world. Do you have any recommendations for this?
HW Payne

Neil says...
I remember PST files back in the day. There were dedicated file readers that would let you directly read individual emails and so on, so I imagine there's very likely something that can import or convert them. In fact, *Outlook* offers export options for *Notes,* I believe, in *Notes:* File > Open & Export > Import/Export > Export To A File, and you can save them as a CSV file. *Joplin* (**https://joplinapp.org**), which we're looking at next issue, has a plugin that supports CSV import. I haven't tried it but this is the theory – if you can get them into *Evernote* or *OneNote*, things look more straightforward with *Joplin.* There are many other note programs, too: *Standard Note, Simple Note, Zim-wiki* and more.

Not spam

Does Ubuntu have any email filters in the Snap Store or even possibly any repos on GitHub you might already know about or use? Not necessarily one that's made specifically for *Thunderbird* only but one that I can use across the whole board on *Hotmail, Gmail* and so forth. I'm just trying to combat unwanted emails from reaching my inbox or spam box instead of having to manually clean up those emails.
Jason Long

Neil says...
I think you're misunderstanding how spam filters and email in general works. Online email providers such as *Gmail* and *Hotmail* store and process your email on their own servers, so if you're using a browser to access them, what you're asking for isn't possible. You can opt to download email from those systems to your desktop using *Thunderbird* or other local email apps, at which point you can use *Thunderbird's* spam filter or another plugin, but a system-wide filter isn't really a thing. LXF



Joplin will get a good going-over next issue.

» LETTER OF THE MONTH

An irrigating problem

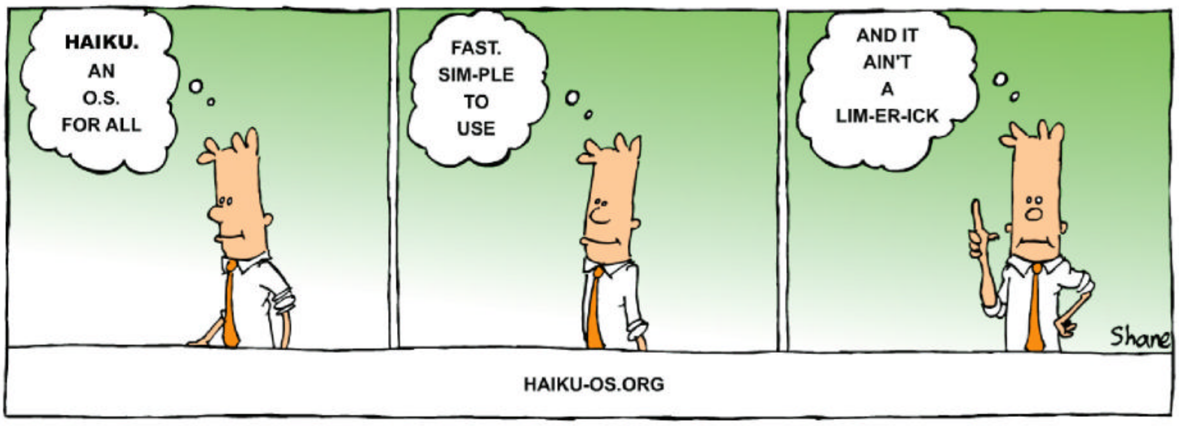
I have been working on software to reduce the water usage of irrigation systems. My goal was to calculate the water consumption based on the plants that need to be irrigated. I also wanted to include rain data and the temperature effect on plant growth.

A number of universities have been studying and publishing how temperature effects plant growth for a variety of plant sorts and I included this data in the latest calculations.

The result is a program that runs on a Raspberry Pi that greatly reduces the water usage for lawns and gardens. I managed to reduce the water costs by around 60% compared to my old conventional irrigation computer.

I am contacting you because it would be nice if this software gets attention now that it is finished. A wide use of this system would help to reduce water consumption and it would save the owners a lot of money. The link for the project is **https://gitlab.com/jtdaling/.**
Jantinus Daling

Neil says...
Another fantastic reader-created project. It amazes me how talented you lot are! We'll do our best to give it some publicity but hopefully the irrigation community (is that a thing?) will be bursting with enthusiasm and overflowing with ideas! Good luck with this!



shane_collinge@yahoo.com

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

» PLUS: Exclusive access[1] to the *Linux Format* subs area!

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

» CHOOSE YOUR PACKAGE!

A year's worth of issues. One payment every six months. Simple.

OUTSIDE THE UK? Turn to **page 65** for more great subscriber deals!

ANNUAL PRINT EDITION

SAVE! 20%

Only £33.50 every 6 months

13 issues of *Linux Format* in print over 12 months and **save 20%**!

ANNUAL PRINT + DIGITAL EDITION

SAVE! 47%

Only £45.99 every 6 months

13 issues of *Linux Format* over 12 months in print and digital, and **save 47%**!

DIGITAL EDITION

SAVE! 39%

Only £27.50 every 6 months

13 issues of digital *Linux Format* over 12 months, and **save 39%**!

*Terms and conditions: **Offer closes 31/7/2023**. Offer open to new UK subscribers only. Pricing is guaranteed for the first 12 months and we will notify you in advance of any price changes. Please allow up to six weeks for delivery of your first subscription issue (up to eight weeks overseas). Your gift will be delivered separately within 60 days after your first payment has cleared. Gifts only available to subscribers on the UK mainland. Colour may vary from that shown. Gift not available with a digital subscription. The full subscription rate is for 12 months (13 issues) and includes postage and packaging. If the magazine ordered changes frequency per annum, we will honour the number of issues paid for, not the term of the subscription. For full terms and conditions, visit www.magazinesdirect.com/terms. For enquiries please call: +44 (0) 330 333 1113. Lines are open Monday-Friday 9am-5pm UK time or email: help@magazinesdirect.com. Calls to 0330 numbers will be charged at no more than a national landline call, and may be included in your phone provider's call bundle.

[1] Only available to www.magazinesdirect.com subscribers

EXPLORE THE POSSIBILITIES THAT LINUX HAS TO OFFER

From open-source software to coding masterclasses and Raspberry Pi projects, get the most from your machine with this exciting array of expert tutorials, guides and advice from the minds behind Linux Format magazine.

NEW
FROM THE MAKERS OF THE NUMBER ONE LINUX MAGAZINE
ULTIMATE LINUX PROJECTS
» SUPERCHARGE YOUR MACHINE «
INSIDE 132 PAGES OF EXPERT ADVICE FROM LINUX FORMAT
ON SALE NOW
» Mint 20 » Smart home office » Retro games » Livestreaming
» Robotic OS » Safeguard data » Escape Google Photos
ULTIMATE LINUX PROJECTS
ISBN 978-1-80023-886-2
FUTURE
FIRST EDITION

FUTURE

Ordering is easy. Go online at:

magazinesdirect.com

Or get it from selected supermarkets & newsagents

Anycubic Photon Mono 2

Denise Bertacchi likes this blend of quick setup with detailed prints.

SPECS

Build volume: 165x89x 143mm
LCD screen: 6.6-inch monochrome
Light source: Parallel matrix
XY axis resolution: 34 microns
Normal exposure time: 2.5 seconds
Interface: 2.8-inch touch panel
Connectivity: USB Type A 2.0
Machine footprint: 390x 229x235mm
Weight: 4kg

Made specifically with first-time resin users in mind, the Anycubic Photon Mono 2 delivers quality 4K prints with a smaller-sized vat and a petite footprint that requires minimal space. Retailing at a rock bottom price, we've noticed serious cutbacks in the machine's base unit and cover, both of which are plastic rather than metal or acrylic, but the print quality is crisp, speed is good, and the machine is very easy to use, making it one of the best resin 3D printers around.

Everything you need to get started is included, even a manual, plus the usual USB drive with Anycubic's *Photon Workshop* slicing software, a PDF of the manual and two pre-sliced test prints.

Coming mostly assembled, you simply screw in the vat and build plate, and you're ready for calibration. Levelling the build plate is fairly straightforward and only takes a piece of paper and an Allen key.

The Photon Mono 2 packs a lot of detail in its prints. It has the same LighTurbo system you'll find on newer Anycubic machines, which provides smoother details. It's on the smaller side of resin printers, but we could fit six presupported miniatures on the build plate at once.

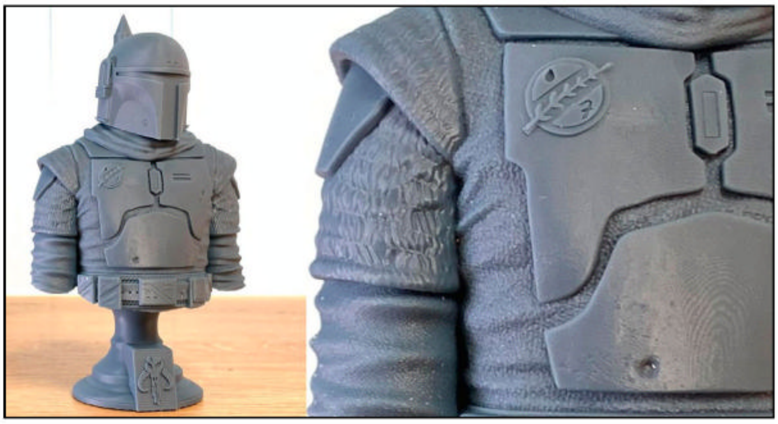
It has a laser-etched build plate that helps prints stick, but things still come off easily enough with the metal scraper. The top of the plate has enough slope to allow most of the resin to drain back into the vat.

Unlike its FDM printer cousins, the Photon Mono 2 reads files from a USB stick rather than a microSD card. The custom slicer *Photon Workshop 3* has improved greatly and we didn't have any problems using it. It handles supports well – automatically and manually. It also has tools for hollowing the model and adding a drainage hole, so you can conserve resin.

We ran a model of Boba Fett at 70% to fit on the plate, with the bust and base printing separately. This printed using Siraya Tech's Fast Navy Resin and a three-second exposure time, taking five hours and 15 seconds for the bust, then another three hours and 10 seconds for the base. The model has incredible details, down to the cracks in the helmet and wrinkles in the sleeve fabric. There are no tell-tale layer lines when viewed with the naked eye and you have to look super-hard to find any with a camera set to close-up.

Anycubic sent a sample of its new High Clear Resin to test with the Mono 2 printer. This is a new formula of resin that can create crystal-clear prints when you follow a simple curing technic. The model printed as normal, using a slightly longer exposure and less curing time. It printed in two hours and 13 seconds, with a four-second layer exposure and no supports. The finished result is as clear as glass where sealed, and frosted where it was just cleaned. LXF

Exceptional print quality hasn't been compromised by cheaper materials.



Bob here really shows off the detail this resin printer is capable of producing.

CREDIT: Anycubic

VERDICT

DEVELOPER: Anycubic
WEB: www.anycubic.com
PRICE: £209

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

A budget 4K resin printer tailor-made for beginners, the Photon Mono 2 delivers great output and easy setup.

» **Rating 9/10**

Intel Core i3-13100F

As Intel reclaims the mid-range, **Paul Alcorn** assesses that the company has lost the low-end.

SPECS

Family: Raptor Lake
Socket: V LGA1700
Process: Intel 7 10nm
Cores (P+E): 4+0
Threads: 8
P-cache: 320KB L1, 5MB L2, 12MB L3 (shared)
P-core: 3.4GHz (4.5GHz boost)
Mem max: 128GB, no ECC, two-channel
Mem speed: DDR5 4,800MT/s, DDR4 3,200MT/s
PCIe: v5 and v4 20 lanes
Unlocked: No
GPU: N/A
Processor base power: 60W
Maximum turbo power: 89W

The Intel Core i3-13100F comes to market with big shoes to fill; its predecessor, the Core i3-12100F, built a reputation as the best budget CPU. Intel is obviously looking to repeat that success, but perhaps the Core i3-13100F follows in the footsteps of its predecessor too closely — the company repurposed the previous-gen design for the Core i3-13100, so it has the same four P-cores paired with an ever-so-slightly-improved 200MHz-higher boost clock of 4.5GHz.

As such, outside of some microcode tuning, the 13100 is identical to the 12100. That is evident from the L2 cache capacity, which weighs in at 1.25MB per core for the 13100. In contrast, the truly new Raptor Cove cores come with 2MB per core.

Like its predecessor, the Core i3-13100F (the F lacks an iGPU) comes with four performance cores and eight threads, but no E-cores for background tasks. Intel sprinkled on 200MHz of extra boost frequency, bringing it to 4.5GHz, and 100MHz higher base clock, taking it to 4.3GHz. The clock speed adjustments are all that Intel has to justify the step up to 13th-gen branding and the much higher price tag, while AMD's Ryzen 5 5600 has seen substantial price reductions that change the circa-£130 landscape entirely.

Picking up speed

If you're looking for a gaming processor for around £100, the £120 Core i3-13100F is 8% faster than the £95 Ryzen 5 5500, giving it the win. The Ryzen 5 5500 is also limited to the PCIe 3.0 interface, while the 13100F supports the faster PCIe 4.0 and 5.0 interfaces, giving it a big drive speed boost.

However, if you're looking for a chip with a little more heft in productivity workloads, or if you can't find a Core i3 F-series model in stock, you'll have to jump up to the £130 to £150 price range. The £130 Ryzen 5 5600 is 8% faster than the Core i3-13100 in 1080p gaming, yet costs $10 less, making it an easy recommendation even before we take its other advantages into account. The £120 Core i3-12100 remains a viable alternative here if you absolutely must have an iGPU (the Ryzen model doesn't have one), but the Ryzen 5 5600 holds the lead in gaming with a discrete GPU and is far more performant in productivity workloads.

Speaking of which, both the Ryzen 5 5600 and 5500 offer substantially more performance in multithreaded productivity applications (30% and 20%, respectively) than the Core i3-13100 and 12100. We think Ryzen's big advantage in threaded workloads will be far more noticeable than Intel's smaller advantage in single-threaded work.

The PCIe v5 abilities are the most tempting aspect of this processor.

The Ryzen processors leverage the plentiful and affordable AM4 motherboard ecosystem, and like the Core i3 chips, support DDR4 memory. Meanwhile, the Intel chips are confined to 600 and 700-series chipsets, so AMD has the platform pricing advantage — you can find incredible deals for AM4 motherboards. Be aware that you lose support for PCIe 5.0 and DDR5 if you go with the Ryzen 5 5600, but we don't think those interfaces are as important with this class of chip. Both the Intel and AMD chips come with bundled coolers that are sufficient for normal operation.

The £150 Core i3-13100 is overpriced for a quad-core in 2023. Intel's decision to use the same design for the 13100F and merely increase boost clocks by 200MHz doesn't do enough to deliver any tangible increase in value. The minor improvements we saw in our testing certainly aren't enough to justify the Core i3-13100's higher price tag over the prior-gen model.

The less expensive £120 Core i3-13100F makes more sense, but it faces a stiff challenge from its predecessor. If you're looking to step up a pricing tier, the Ryzen 5 5600 is the best of the bunch in both gaming and productivity work. LXF

VERDICT

DEVELOPER: Intel
WEB: www.intel.com
PRICE: £120

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

A rehash of the previous-gen model, with slightly higher pricing and clock speeds, so performance gains are limited.

» **Rating 7/10**

Fedora 38

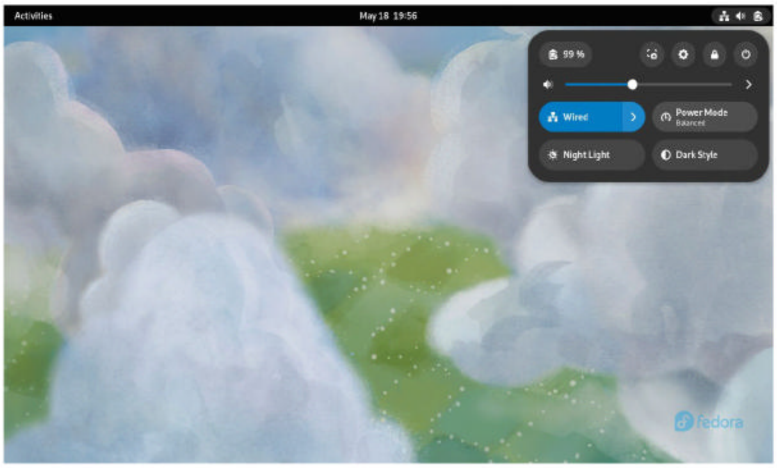
Nate Drake took the latest Fedora for a test drive and was blown away by the vast array of new spins and features.

IN BRIEF

Like Indiana Jones's hat, Fedora is cool, adventurous and easy to don. There's a kaleidoscope of spins combined with easy setup, bleeding-edge features and a huge selection of software.

SPECS

CPU: 2GHz dual-core
Mem: 2GB
HDD: 15GB
Builds: x86_64, AArch64



The revamped Gnome Quick Settings display more information about individual features and enable you to connect or disconnect paired Bluetooth devices.

Apart from being the chosen hat of everyone's favourite adventurer, Indiana Jones, Fedora is the upstream source of the Red Hat Linux Project. This puts it on the bleeding edge of innovation when it comes to software packages, so we were eager to see if Fedora 38 lived up to its predecessors. This latest version was announced on the Fedora blog a week early, which came as rather a surprise to the Linux community, which is used to a more sedentary development cycle.

If you've already decided on the Workstation (desktop) version of Fedora, you're still spoiled for choice when it comes to desktop environments. Apart from the default Gnome (pictured), there's now an official spin of the Budgie, KDE, Xfce, LXDE, MATE and Cinnamon environments.

Once you've chosen, you can either download a special image writer tool to copy your chosen flavour to USB or just choose the ISO. We were pleased to see the Gnome flavour of Fedora 38 was a very trim 2GB.

Once you've had enough of test-piloting Fedora in live mode, you can fire up the *Anaconda* installer, which can automatically partition your drive and encrypt it.

On first boot, Fedora launches a setup guide that lets you choose whether to enable location services and error reporting. You can also choose to enable third-party repos to install proprietary software such as drivers, as well as an unfiltered view of apps on Flathub. Finally, you're asked to set a username and password.

This done, you'll find an impressive overhaul of the interface thanks to Gnome 44. The cloudy default wallpaper may not be to everyone's taste but there's a number of alternatives in Gnome's settings.

While we're on the subject, the new Gnome lock screen displays a blurred version of the desktop in the background. The Quick Settings menu has also been overhauled, now showing both the name of a setting with its status below, such as 'Power Mode – Balanced'. You can also view and close background apps here.

Although the Quick Settings menu has previously displayed Bluetooth, you can now more easily disconnect and reconnect to previously paired devices.

The file manager now also supports icon view and previewing of image files. You can also now navigate folders more easily via the list view. (You need to enable this in the file manager Preferences if you want to use this feature.)

Fedora 38 uses version 6.2 of the Linux kernel, which has hugely improved hardware support and bug fixes. Pre-installed apps include the latest versions of *Firefox, LibreOffice* and *RhythmBox.*

You can install more via Gnome Software. You may want to look into this on first install as Fedora still has no bundled email client.

The lightweight *microdnf* package manager has also now been replaced with *dnf5,* which has a radically smaller memory footprint. You can also still disable individual software repositories via Gnome's settings.

Fedora 38 has stricter rules about how long running services can block shutdown. In theory this should help devices running it to power off faster and efficiently.

The Fedora Project states the four main foundations of what it does as "Freedom, Friends, Features and First". It's precisely its friendship and sponsorship with Red Hat that allows it to include the latest innovations in its OS, with amazing features such as those in the new Gnome 44 desktop. We're also pleased to see its devotion to freedom in software now extends to giving users the choice of whether to use third-party repos and programs instead of filtering content for them. LXF

VERDICT

DEVELOPER: Fedora Project
WEB: https://fedoraproject.org
LICENCE: GPL

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

Fedora has outdone itself again with a smooth setup and innovative features, not to mention all the available spins.

» **Rating 9/10**

EuroLinux 9.2

Nate Drake is on the continent this month, taking a stroll through the natural beauty of EuroLinux.

IN BRIEF

Does the world need another RHEL-based distro? In the case of EuroLinux, the answer is a resounding yes. It's easy to set up, visually stunning and supported through to 2032.

SPECS

CPU: 1GHz
Mem: 4GB (1.5GB per CPU recommended)
HDD: 10GB (20GB recommended)
Builds: x86_64, AArch64 (ARM64), i686

EuroLinux belongs to a select group, as it's one of the six major Linux distros developed in Poland.

It is built upon the foundation of RHEL (Red Hat Enterprise Linux) source code. The EuroLinux team claims this guarantees compatibility with RHEL, Oracle Linux, CentOS, AlmaLinux and Rocky Linux.

From reciting this shopping list of distros, it seems the team is aware there's no shortage of options if RHEL-based Linux is your jam. Rocky Linux, for instance, is also based on Red Hat and uses the Gnome desktop environment.

Luckily, EuroLinux has an answer for why the world needs another RHEL-based distro: "EuroLinux was developed in response to unsatisfied needs of the market related to the quality of technical support, unfavourable licensing, high cost of subscriptions and the lack of flexibility of the leading producers of Enterprise Linux distros."

Both the server and desktop distros are available free of charge under the GNU General Public Licence. This means in theory you're free to modify and distribute it as you wish. EuroLinux makes this especially easy through its *EuroLinux Gaia* tool.

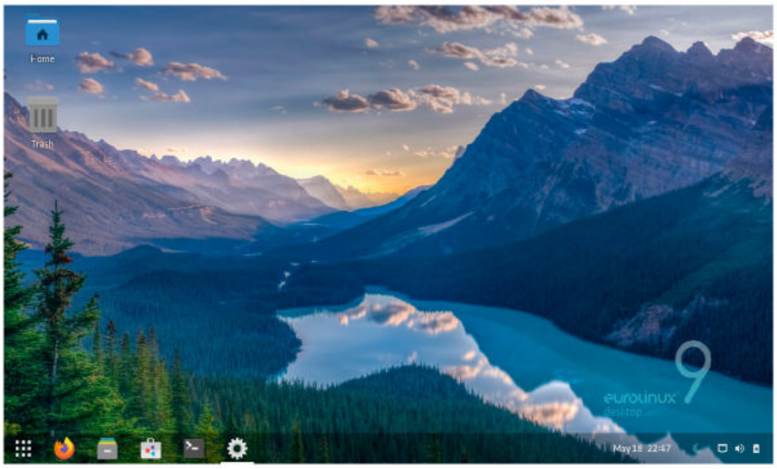
However, if you want to use proprietary tools such as *Gaia*, benefit from tech support, or receive installation assistance and intermediate packages, you need a paid subscription. EuroLinux offers three tiers of support but when we tried to find a pricing page, we were invited to fill in a contact form for a quote.

We focused on the latest free desktop version of EuroLinux (9.2) and came away very impressed, even without the benefit of paid tech support. At 4GB, the ISO took a while to download but the *Calamares* installer was a breeze. The support pages do have a basic setup guide, which helpfully reminded us to tick Make Administrator when creating a user account. You can also encrypt the hard drive for extra security.

The Gnome desktop's default EuroLinux wallpaper is very picturesque, but you can choose others.

The default apps are well chosen – web browsing is provided by the latest ESR (Extended Support Release) of *Firefox*. You can access email via *Geary* or *Evolution*. EuroLinux also comes with version 7.1.8 of *LibreOffice*. If you're noticing a theme, it seems these packages have been selected for stability, so don't represent the bleeding-edge versions currently available. On the plus side, EuroLinux 9 will be supported until 2032.

When we fired up the Gnome software centre, we noticed a slight quirk when trying to install desktop utility *Damask*, which was available as a Flatpak. Gnome Software said it wasn't supported but it installed readily enough and opened without issue. Upon switching to the command line, we had no issue installing the *gnome-tweaks* tool using the *dnf* package manager.



Take a gander at EuroLinux's stunning vistas on Gnome. Browse with Firefox or use Gnome Software to install more apps.

The latest version of EuroLinux (9.2) comes with numerous other improvements under the hood. There are security updates to OpenSSL and the SELinux user space packages.

Developers will be pleased to hear that EuroLinux is also now bundled with Python 3.11, as well as the **java-11-openjdk**, **java-17-openjdk** and **java-1.8.0-openjdk** packages. You can read the full release notes at **https://bit.ly/lxf304euro**.

If Red Hat is your thing and you don't want to pay huge amounts for a licence, EuroLinux may be the distro for you. It's certainly designed for stability and the interface is effortless. Just remember that if you want to install the latest versions of applications, you need to be comfortable with the command line. LXF

VERDICT

DEVELOPER: EuroLinux
WEB: https://en.euro-linux.com
LICENCE: GNU GPL

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

EuroLinux could easily go 12 rounds with Rocky. It's slick to install, supported for years and provides a clear interface.

» **Rating 8/10**

Peropesis 2.1

Nate Drake grapples with the ultra-minimalist Peropesis. Will he relive Linux's glory days, recover a corrupted system or have a breakdown?

IN BRIEF

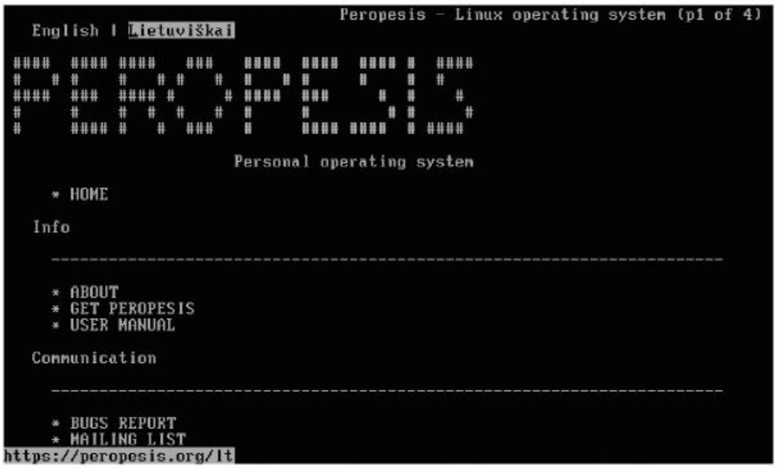
Peropesis has a tiny footprint and loads into RAM in seconds. Next, master the Linux command line and compile your own programs – if you have the time and patience.

SPECS

CPU: 1GHz
Mem: 256MB
HDD: N/A (Live only)
Builds: x86_64

Not to be confused with the capital of the First Persian Empire (that's Persepolis), the developers of Peropesis ('personal operating system') say it is "small-scale, minimalist, command-line-based Linux".

How small scale? This distro's ISO weighs in at just 205MB and is designed to be run entirely in live mode. The helpful introductory manual on the Peropesis website says the distro is a good way to familiarise yourself with the command line, as well as programs commonly used in Linux.



```
                                        Peropesis - Linux operating system (p1 of 4)
  English | Lietuviškai

                          Personal operating system

    * HOME

  Info
    ------------------------------------------------------------------------

    * ABOUT
    * GET PEROPESIS
    * USER MANUAL

  Communication
    ------------------------------------------------------------------------

    * BUGS REPORT
    * MAILING LIST
https://peropesis.org/lt
```

You can use Peropesis's text-based Links browser to load websites. There are no graphic options but it's surprisingly simple to navigate.

It also points out that this system can be quickly booted in live mode and is compatible with UEFI and BIOS motherboards. That makes Peropesis perfect for mounting hard drives of faulty systems.

A cynic might argue you could do exactly the same by live-booting the latest version of Ubuntu via the terminal and enjoy a quick game of *Mines* on the desktop as your drives are being rebuilt.

Still, Peropesis has a certain charm. It's entirely distribution-independent, having been written from scratch with open source code. It's also the one and only Lithuanian Linux distro we've reviewed to date. The online manual makes it far less daunting than other command-line-only distros we've encountered.

Your first hurdle involves simply entering the `root` username to log in – no password required.

After this you can navigate the file structure or run one of the few built-in apps. Peropesis includes the *Links* text-only web browser for visiting web pages. Sadly, this isn't to be confused with the *Links2* or *Lynx* browsers, which have limited image support via X11, but it's surprisingly easy to navigate pages via text only.

Making good on the earlier claim, the manual also covers how to use Peropesis to mount an external USB drive and use `chroot` to recover a faulty Linux OS.

On login, you may also notice the message saying 'No New Mail'. This is thanks to the *S-nail* mail client. As this again can only be accessed via the command line, you need to edit the **~/.mailrc** yourself to add your credentials. Peropesis supports the `cat`, `nano` and `ed` commands for viewing and editing files.

You can also navigate directories via `pwd`, `ls` and `cd`. We used these to list the contents of **/bin** to see what other programs come preinstalled. Luckily, Peropesis supports both the `man` command and `--help` flag if you see anything unfamiliar.

If you want to install any further programs, there's no package manager. This means you have to use `wget` to download the tarball of your chosen program, extract it via the command line and run `configure` and `install`. This will probably turn off Linux newcomers who are used to a colourful software centre.

Still, you may not have much of a choice if you want to boot the ISO on a system that uses proprietary firmware. The Peropesis manual cautions that the distro only includes a small amount of firmware but also shows you how to view system logs to check that the correct software is installed. The Peropesis forum (**https://peropesis.org/forum**) has a lively discussion on this topic, with users suggesting a more visually friendly package manager like *Pkgsrc* may be in order.

Visit **https://peropesis.org/bundle/** to see a full list of updates to the latest version of Peropesis. These include support for the Perl programming language (Python is notably absent, however). There are also new versions of *Autoconf, Automake* and *GCC* to make compiling your own programs easier.

Officially Peropesis is in the open beta development stage, but we plan to keep an eye on it in future. LXF

VERDICT

DEVELOPER: Gediminas Mockus and the Peropesis team **WEB:** https://peropesis.org
LICENCE: CC BY-SA 4.0

FEATURES	3/10	EASE OF USE	6/10
PERFORMANCE	9/10	DOCUMENTATION	9/10

Peropesis introduces users to commands and the file structure of the Linux OS. It also boots in record time.

» **Rating 7/10**

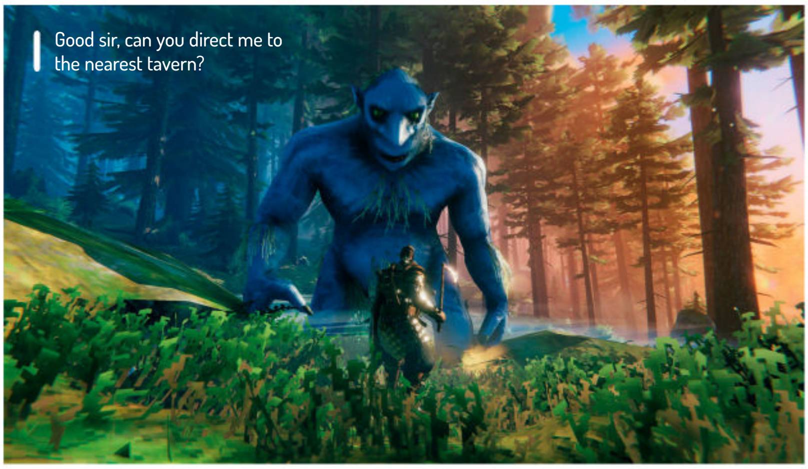
Valheim

Management are delirious to learn that **Christopher Livingston** can keep working long after his death and will take draugr entrails as payment!

SPECS

Minimum
OS: 64-bit
CPU: 2.6GHz dual-core 64-bit
Mem: 8GB
GPU: GeForce GTX 950, Radeon HD 7970
HDD: 1GB

Recommended
OS: 64-bit
CPU: 3GHz Core i5, 3GHz Ryzen 5
Mem: 16GB
GPU: GeForce GTX 1060, Radeon RX 580
HDD: 1GB



Good sir, can you direct me to the nearest tavern?

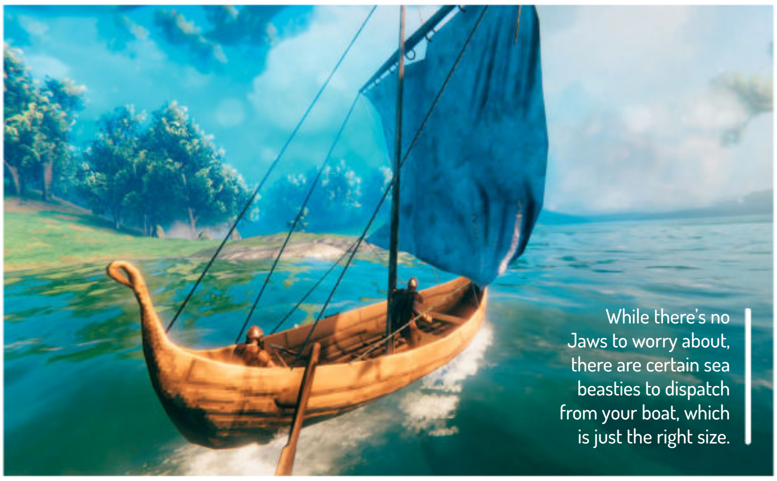
This is an Early Access preview of a game that's still in active development and updating. Originally released in June 2020, we feel safe casting an eye over where *Valheim* has been, where it is and where it's going…

Our first trip across the ocean was on a tiny wooden raft, holding a torch nervously, peering through the pitch-black night. It makes you feel intensely vulnerable. Having never left the starter island before, we have no idea what's waiting out there in *Valheim's* massive procedurally generated world. After a long, tense night of sailing, finally setting foot on a new continent, we immediately discover what looks like a village. That's a surprise – we hadn't know there were villages in *Valheim.* The village is full of draugrs. We didn't know there were draugrs, either. This game is full of surprises. Don't ask about the trees…

A mob of undead warriors bash us with axes and bombard us with arrows. Fleeing home in misery, we have little to show for hours of exploration save for badly degraded weapons and armour, and a few draugr entrails. We decide never to go back there. Ever. But the discovery of draugr intestines has given us a recipe for sausages, so we stuff the entrails with boar meat and flavour them with thistle. Then eat them, eyes widening as the health bar grows to twice the size it's ever been.

In *Valheim*, you're a dead Viking warrior. Your soul has been deposited in the afterlife so you can battle the enemies of Odin, powerful creatures such as a towering giant made from tree trunks and a toxic swamp blob that emits great clouds of poison.

But before you can do Odin's work, you've got to do dozens of hours of your own labour: building a home, making weapons and gear, levelling up skills, unlocking crafting recipes, and slowly exploring deeper and deeper into the huge, dangerous world. It may not sound all that different from other open-world survival sandboxes, but *Valheim* is an utterly engrossing experience that blends thoughtfully designed survival systems with exciting RPG-like adventures, where each small nugget of progress sets the stage for the next.



While there's no Jaws to worry about, there are certain sea beasties to dispatch from your boat, which is just the right size.

Odin's blood

The sausages are a good example. Unlike most survival games, you won't starve to death in *Valheim* if you don't eat, but you absolutely need to eat. The right foods dramatically boost your tiny health bar and increase your stamina, so you won't get far without spending some time in the kitchen. And a whole lot of deaths along the way. There's not so much a difficulty curve to *Valheim* as there are towering, razor-sharp difficulty spikes. That feels frustrating initially, but eventually, and weirdly, it becomes encouraging. Just setting foot somewhere you're not ready for, like that draugr village or a swamp crypt or a frigid mountainside, can brutally punish you, but also give you new goals and a tantalising glimpse of future possibilities.

When first discovering a new biome, the Plains, we had roughly one second to admire the view and swelling music before a deathsquito buzzed across the screen and into our side, taking more than half our health away with one jab. We fled immediately, though we managed to kill the insect, gaining a needle, which gave us the crafting recipe for a deadlier type of arrow.

Strength in numbers

We've split time in *Valheim* between solo play and adventuring on a server with some others, and while they're both rewarding, playing with friends gives *Valheim* a wonderful communal feeling. We've built a small settlement with several buildings, we share resources and discoveries, take on boss fights together, and help each other out when it comes to personal missions and goals.

One of those missions was a rescue and recovery operation. It was a long sail, made more complicated when a sea serpent, the first we'd ever encountered, attacked us in the middle of the night. While we shot the creature with flaming arrows, Steven took us to shore, fearing our boat would be destroyed. Once on land, we were mobbed by growling greydwarfs while the serpent continued attacking our ship. We finally, frantically, dealt with both threats and set off again, only to realise we didn't have enough resources to build a fast-travel portal.

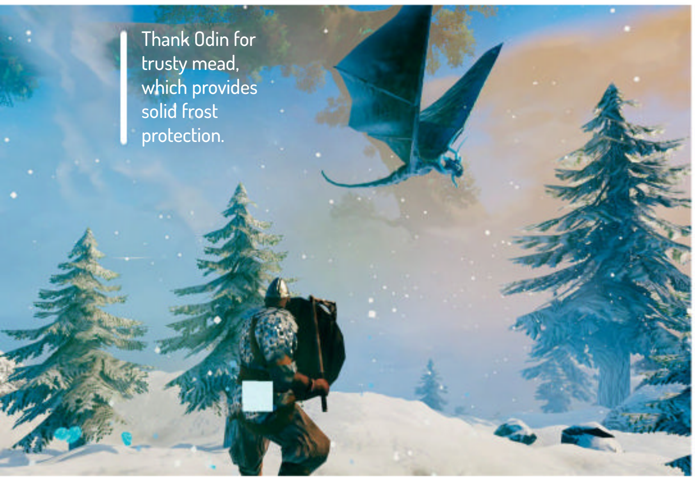
So, we had to make another stop to collect wood in the darkness of night while we built a workbench to repair the damage the serpent had done to the ship. Finally, we reached the area where the boat was lost. After another mad scramble we killed a goblin, recovered the gear, and had a delightful and peaceful sail back home, each in our own boats. It was a genuinely exciting adventure, with one extra bonus: we now had serpent meat, which gave us the recipe for serpent stew, a fantastic new health and stamina-boosting food.

Bosses, however, provide some structure to the otherwise open-ended adventure. Finding them takes a ton of exploration, because only certain runestones show their location on your map – and a boss might wind up several continents away from your starting island. Just reaching a boss with the resources you need to summon them is an adventure in itself. And the boss battles are long, challenging bouts accompanied by music and effects that really make you feel as though you're in a dramatic showdown with angry gods. Each boss drops an item you need to begin the long process of preparing to take down the next one.

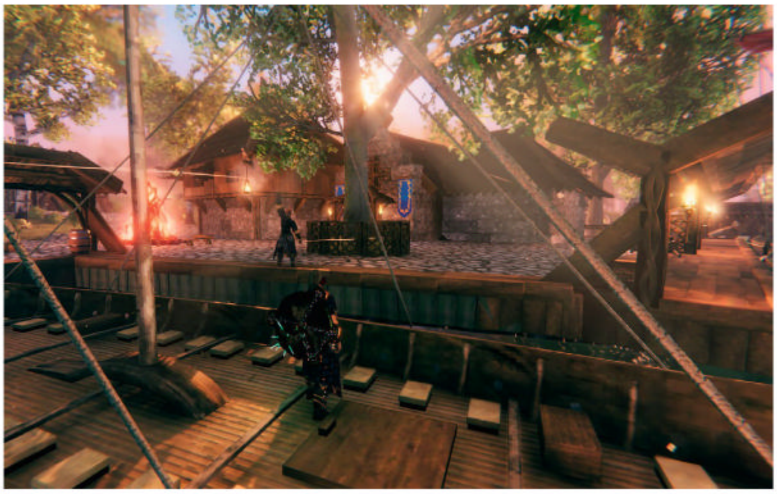
Tree of life

The biggest update to *Valheim* dropped at the end of 2022 as the Mistlands pack, which went way beyond adding just a new biome type. It packed an entirely new gameplay system, an interesting new type of NPC, several unusual crafting stations and lots more.

There's always been magic running through the world of *Valheim*, but now you can wield it directly. Thanks to a mystical new resource called Eitr, you can imbue yourself with magic power and build a new workbench to create several types of magic staff. One staff can hurl fireballs, another can deal ice-blasts, and there's a staff that encases you and any nearby allies in a protective force bubble. Along with the staffs, there's a craftable magic skull for summoning a loyal skeleton to fight at your side.



Thank Odin for trusty mead, which provides solid frost protection.



A village! Buildings can be destroyed for resources and built from scratch to create your own settlements.

Just don't assume you're instantly gonna be the Viking version of Gandalf. This is a magic system, not a magic button. Harvesting and refining Eitr is quite a process, as is gathering what you need to build magic-related crafting stations. And without Eitr flowing in your body, your fancy new staff might as well be a twig.

In such a hostile world, it doesn't take much for things to feel more friendly, even if that friendliness is simple neutrality. You've already met one Norse dwarf, Haldor, the vendor of *Valheim*. But in the Mistlands you'll find other dwarf clans, called Dvergr, and unlike most things in *Valheim,* they don't want to kill you.

Since we first set foot in *Valheim,* we've been staring up at the giant tree, Yggdrasil, that stretches out over the Viking land of the dead. We've just never been able to reach it. As it turns out, the mighty tree's roots are sunk deep in the Mistlands – they're not just there for decoration. You can craft a new tool to extract a precious new resource from the roots of Yggdrasil, which is key to fully unlocking that new magic system.

We knew the new biome would have a new boss, but this boss has a lair rather than an altar in the overworld. Sealed by Dvergr long ago, once you've found the lair you need to craft a special key just to open that intimidating door. A creature so foul it's been locked away in a vault? Sounds like you should leave that door closed…

And there's chickens! In *Valheim!* You can hatch 'em, grow 'em, and farm 'em at your base. They're not just cute, they're also delicious. Now all you have to do is find your first egg – somewhere. LXF

VERDICT

DEVELOPER: Coffee Stain Publishing
WEB: www.valheimgame.com
PRICE: £15.49

GAMEPLAY	9/10	LONGEVITY	10/10
GRAPHICS	7/10	VALUE	9/10

Despite being Early Access, the gorgeous and engaging Valheim feels refined and satisfying as it is right now.

» **Rating 9/10**

Roundup

Ubuntu Studio » Kubuntu » Edubuntu » Ubuntu MATE » Lubuntu

Michael Reed has been handed so many flavours and spins over the years that he feels like he works on a milkshake stand.

Ubuntu spins

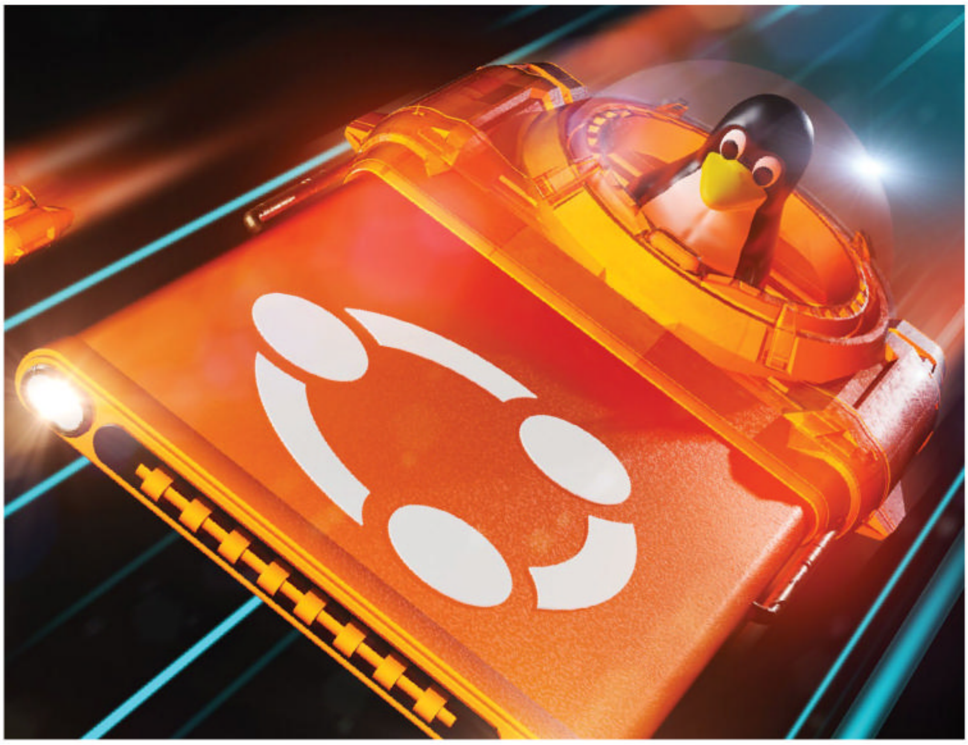
With the release of Ubuntu 23.04 still fairly recent, **Michael Reed** examines the differences between the official spins of that distribution.

HOW WE TESTED...

For this assignment, we installed each distribution and began playing around with them, trying to picture how they would look to both newcomers and people who already have some Linux experience. We carried out some basic tasks and loaded up most of the important apps. We updated the systems and tried installing some software using the GUI package management tools on offer.

This time around, we're not searching for a low-end solution, but we did make some notes about how efficient each distribution is. As usual, we used the `df -h` command to display the disk usage of an updated fresh installation and `free -h` to find out how much memory was in use upon a fresh boot to a desktop with a terminal window open. It's sometimes subjective, but we keep a constant lookout for anything that feels slow.

In this case, no notable problems, such as crashes or slowdowns, showed up when managing software or using the systems.



CREDIT: Magictorch

To some of us, they are spins or editions, but Ubuntu calls them flavours. We're going to take a look at five of the official flavours of Ubuntu 23.04, and we can tell you now, each one is worth considering and has its own strengths.

Kubuntu offers a KDE 4 experience with great looks and a polished desktop, along with all the basic applications you're likely to need to get started. It's modern, polished Linux desktop computing at its best. Lubuntu uses the *Qt*-based LXQt desktop environment with a combination of standard desktop applications and lightweight tools. Ubuntu MATE delivers a fairly conventional desktop environment that's well suited for either the office or home.

Ubuntu Studio is a specialised distro that combines a customised KDE desktop with an extensive selection of applications, tools and plugins that cover areas such as music, audio, graphic design and video editing, all preconfigured and ready to go. It's the ideal starting point for building a studio computer. Edubuntu is also specialised but for educational use. To achieve this, it offers a Gnome desktop along with a large selection of apps and smaller tools that are ideally suited to classroom use or to make a fantastic child-friendly computer setup.

The applications

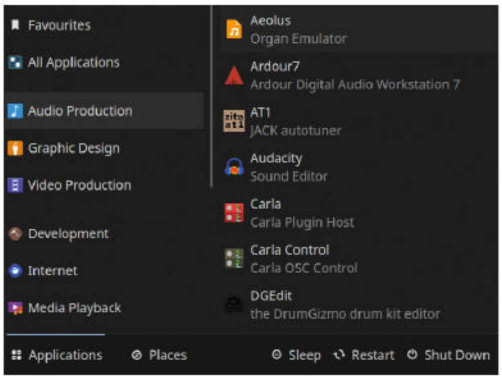
A distro gets a lot of its personality from the default selection of apps.

Alongside the choice of desktop environment, the default application selection is the biggest distinguishing feature when comparing a set of Ubuntu flavours. Indeed, we hope for a good match between the applications and the desktop environment.

As Kubuntu is based upon KDE, it's no surprise that many of the tools are connected with that environment. However, where sensible, generic apps have been favoured. The inclusion of web browser *Firefox* and office suite *LibreOffice* are examples of this. We're always happy to see KDE's file manager, *Dolphin,* as it's one of the best. KDE apps such as text editor *Kate,* music collection manager *Elise* and media player *Haruna* round out the selection, so for most users, Kubuntu won't need much adding to it to cover basic use. The use of bigger apps that aren't part of KDE isn't too jarring and fits quite well alongside the KDE apps.

Ubuntu Studio is based on KDE as well, and it comes with a full complement of standard applications such as *Firefox.* Part of the concept behind the distro is that it also comes with a complete set of installed and configured media applications, and the installed selection is huge.

Apart from the music applications such as *Ardour,* there is a full set of installed plugins that cover every area of music and audio production. There are also applications that cover other areas of media production, such as graphics editing, drawing and 3D graphics. Some duplication of features is unavoidable with such a comprehensive selection. For example, some people might not need *GIMP* if they have *Krita* installed. Furthermore, not every musician needs a 3D graphics application like *Blender*. In many cases, configuring media applications can be a hassle, so the fact that it's already done is worth quite a lot, particularly for beginners.



Ubuntu Studio comes with a lot of weighty applications and smaller utilities, aimed at media creation, already installed. Thankfully, they are categorised for easy browsing.

Another specialist distribution, Edubuntu takes a similar approach with a full complement of both standard and specialised applications. It's absolutely packed with small tools that cover areas such as science, geography and mathematics, alongside larger applications such as diagram creator *Dia* that would be useful in a classroom.

Lubuntu covers the basics with *LibreOffice* and *Firefox.* Ubuntu MATE takes a similar tack with a selection of standard applications that cover typical office and home use.

VERDICT

UBUNTU STUDIO	9/10	UBUNTU MATE	7/10
KUBUNTU	8/10	LUBUNTU	7/10
EDUBUNTU	9/10		

All of the distros can cover basic use, but Edubuntu and Ubuntu Studio offer masses of specialised applications on top of that.

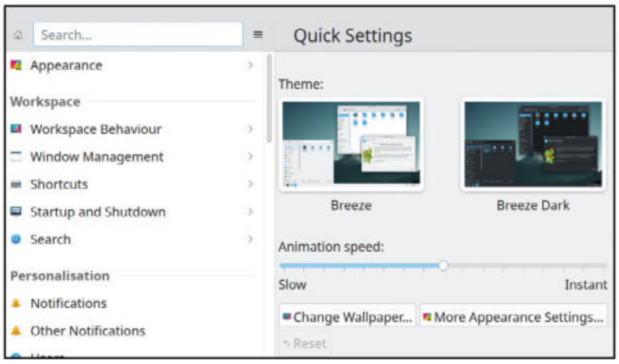
Configuration tools

Hopefully, the settings cover all of the main areas of the system and its appearance.

Ubuntu Studio and Kubuntu are both KDE-based distros, and KDE now groups all of the configuration into a single application, which is searchable. Having a central location for the settings doesn't come with any real disadvantages. The configuration tools cover all the common areas of the interface along with system settings in areas such as network setup. The various settings show up in the launcher's search results as well.

LXQt-based Lubuntu groups the settings together in the LXQt *Settings* app, but there are settings applets that exist outside of this, meaning that the entire collection isn't searchable. One way or another, it should be possible to alter any system or appearance settings that you need to, even if the tools are neither fully integrated nor consistent in appearance like the KDE ones.

The settings for Edubuntu can be accessed in two ways. Firstly, Gnome's *Settings* app gathers all settings in one place, and it's searchable. All the settings pages are also present in Gnome's search-based launcher. So, if you searched for 'resolution' in the launcher, it would list Display Settings and launching that from the list would launch the *Settings* app open on the correct tab.



The Gnome settings offer uniformity but the KDE settings have more colour and variety. Other than that, Gnome and KDE are fairly even in this area.

MATE's *Control Center* groups all of the settings applets together. It is searchable, but the search is limited to the applets and not the settings themselves, and the applets aren't integrated into *Control Center*.

VERDICT

UBUNTU STUDIO	9/10	UBUNTU MATE	7/10
KUBUNTU	9/10	LUBUNTU	6/10
EDUBUNTU	9/10		

Gnome and KDE are fairly even in this area, but LXQt and MATE would benefit from greater integration and more comprehensive search.

The user interface

Your connection to the system.

All of these desktops can be customised, but we're evaluating the default desktop experience. Evaluating the user interface of a distribution, we expect something that looks good and is easy to understand and use. At this point, we expect the application launcher to be a searchable one, and we appreciate a smooth, mouse-free workflow, from summoning the launcher to locating the correct application to, finally, launching it.

User interface looks aren't just about prettiness. We like something that's consistent and clear. Most machines that are in use today have a few pixels to spare, so some use of space is welcome. For example, some Linux window managers seem to like to impose tiny window controls and a layout with insufficient visual indication of where a user interface area starts and ends. There's no accounting for taste, but user interfaces with a completely flat look are difficult for new users to understand at a glance.

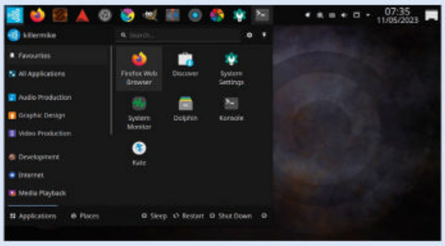
Ubuntu Studio 8/10

Given the use of dark colours, you might not think this is a KDE/*Plasma* desktop.

The desktop uses the standard launcher/switcher panel, but it puts it at the top, and there is a selection of quick-launch icons on the bar as well. The searchable launcher uses categories and some of these expand on the standard defaults with helpful titles such as Audio Production and Graphic Design, which is a nice touch that aids navigation among the many specialised apps that are included.

Compared to other similar distros, we found that we needed to increase the size of the font and the main panel, but these tweaks were relatively easy to carry out.

Although we're used to seeing bright default themes with KDE, the desktop is easy to navigate. Sure enough, when we launched apps such as text editor *Kate* or graphics editor *GIMP,* they also used a dark theme, aiding overall consistency.

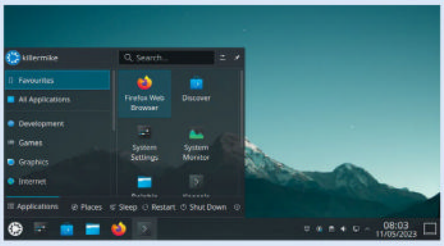


Kubuntu 9/10

Properly configured, KDE 4 (with the *Plasma* window manager) offers about the best balance you're likely to find of a traditional desktop with modern, clean looks and lots of features. It's not the lightest option, but with a reasonably modern computer, that won't make much difference. It's airy, with well-spaced elements and clear lines separating different areas. It's also close enough to traditional desktops that it works in a way that most people will be familiar with. By default, the task switcher is icon-based rather than using the traditional task panels, but it's easily switchable.

The super key launches the application launcher, which is searchable, but it's also tied into the constantly updated desktop search features, meaning that it is a hub for finding documents as well as apps.

KDE 4 is maybe the most polished-looking Linux desktop around right now.



Drive and memory efficiency

Extra efficiency is like having a faster CPU without spending money.

We're not focusing on performance on low-end machines in this *Roundup*, but all of these distributions should be way ahead of Microsoft Windows on such hardware. We've reached a point where even a comprehensive Linux desktop doesn't require a huge chunk of the memory in a computer with only 8GB or even 4GB of RAM at a push.

If you were working with an extremely resource-constrained machine, Lubuntu would be our first choice of distro from this selection. On a fresh boot, only 560MB of RAM is in use. Lubuntu was also the least disk-space-hungry of the flavours on test, only requiring 7.3GB of space on an updated fresh installation.

Kubuntu offers a state-of-the art Linux desktop that looks great, and it proves our point as it only uses 833MB of RAM and takes up 16GB of disk space. We were slightly surprised that Ubuntu MATE also used 16GB of disk space because it's not particularly packed with applications, but it's not that important in most situations. It also used 886MB of RAM, which is perfectly fine for computers with 4GB of memory or more – modern browsers are the real hogs.

Edubuntu is also far from being a resource hog as it only used 962MB of RAM. It used the most disk space of any of the flavours at 19GB, but that's hardly surprising, given how many applications it packs in. Much the same can be said of Ubuntu Studio. This uses 1GB of RAM on fresh boot and 14GB of disk space, neither requirement a big deal on the sort of computer you'd probably use for media creation work.

VERDICT

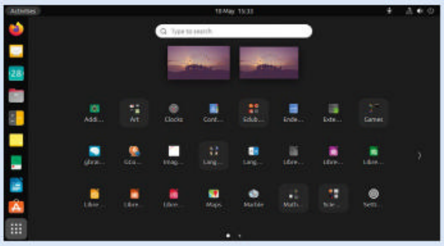
UBUNTU STUDIO	7/10	UBUNTU MATE	7/10
KUBUNTU	8/10	LUBUNTU	9/10
EDUBUNTU	7/10		

None of the distributions are resource hogs or slow in any way. Lubuntu was the lightest overall, though.

Edubuntu 9/10

The layout of Edubuntu is largely the layout of standard Ubuntu with a custom colour scheme. Its Gnome desktop is configured with a vertical dock launcher/switcher that sits against the left-hand side. Single clicks on these icons launch apps and the icon becomes highlighted once the app is running. The layout isn't traditional, but it's so simple that most people can quickly pick it up, an important consideration for an educational system.

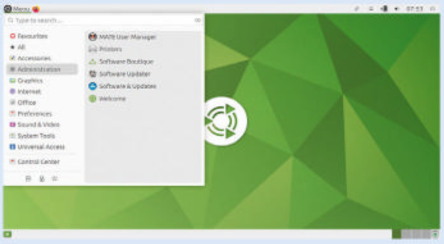
There is a slight issue in that categories are thrown in with apps in the pop-up launcher, following the standard Gnome scheme. In this case, there are many specialised educational apps and although locating and launching apps is fast using the Gnome launcher, it's not quite as good as a traditional one for browsing via categories. This criticism aside, the standard Gnome desktop is easy for adults and kids to pick up, and works well.



Ubuntu MATE 7/10

As it relies on MATE as the desktop environment, this doesn't break much new ground in terms of user interface concepts. What it does offer is familiar and solid. Good use is made of a mild 3D effect that has fallen out of fashion with some other desktops and around the web, and the amount of space around elements is good. Personal preference might be a factor; it is an old-fashioned-looking desktop as it has its origins with the original Gnome desktop, from which it was forked during the 2.x era.

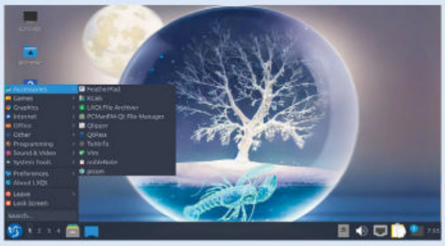
The standard, searchable launcher pops down rather than up because the taskbar is at the top of the screen, and a second bar containing the task switcher runs along the bottom. The advantage of doing it this way is that the taskbar has plenty of room on it for old-fashioned task panels with full window titles on it.



Lubuntu 8/10

Lubuntu is powered by LXQt, a desktop environment that has a look reminiscent of *Qt*, the toolkit that powers it. This means it's neat and clear, and not overly flashy. You certainly don't have many moments of feeling confused as to what does what. Although simple and easy to understand, it has a functional style that never feels bland, and it's nice to see some use of colours in the icons.

This familiarity and simplicity extends to the overall desktop layout, with the launcher/taskbar/status panel located along the bottom of the screen. That launcher is searchable, as we'd expect, although it can only search by item name and not description. It's a crystal-clear desktop that's dark without being drab – it has a look that goes against some of the current UI trends, and we like it. None of the desktops were slow, but LXQt felt like the lightest and fastest.



Package management

There is a variety of front-ends to the Ubuntu package system.

Canonical has famously disabled Flatpak support recently, and each of these distributions is affected by that, but you can enable support by manually installing it, as you would on normal Ubuntu. As these flavours are Ubuntu under the hood, you can also use the standard package installation tools such as *Apt* on the command line along with the Snap packaging system.

KDE's application launcher (as employed by Ubuntu Studio and Kubuntu) is linked to the packaging system, making installation suggestions while you search. Both of those distributions feature a software store called *Discover.* This features applications from both the Ubuntu and Snap repositories with reviews and ratings.

Ubuntu MATE has its own software manager called *Software Boutique,* but in its current state of development, it's not the best store app we've seen. Its database isn't very extensive – although it does contain a mix of open source and proprietary software – and it doesn't feature ratings or reviews. We'd use it to install a basic GUI package manager such as *Synaptic* or a more extensive store such as *Gnome Software.*

Edubuntu comes with *Ubuntu Software Center,* Canonical's front-end to the Snap system. It contains a curated list of apps that are, helpfully, rated and reviewed. You have to install something like *Synaptic* if you want to be able to install smaller apps and system components that aren't in the Snap repository.

Muon is a *Qt*-powered package manager on Kubuntu, Ubuntu Studio and Lubuntu. It's a front-end to Ubuntu package repos, and is similar to the *Synaptic* front-end on GTK-based distros.

VERDICT

UBUNTU STUDIO	8/10	UBUNTU MATE	6/10
KUBUNTU	9/10	LUBUNTU	7/10
EDUBUNTU	7/10		

KDE's application launcher suggestions are useful, and by default Ubuntu Studio and Kubuntu tick the most boxes.

Suitability for switchers

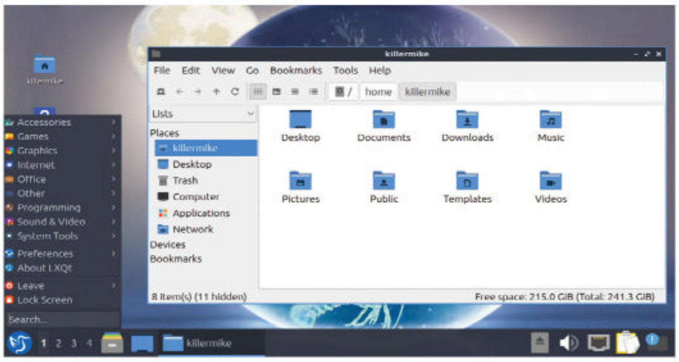
How suitable is this distro for those used to a different OS?

Ubuntu Studio has some unique advantages in this area thanks to its vast selection of installed and configured media applications and plugins. The task switcher at the top of the screen should be fairly easy for Windows users to figure out on first use, and the launcher is fairly standard.

We could say some of the same things about Edubuntu. A newcomer to Linux would be amazed with what it offers out of the box, particularly if they are interested in classroom use or a computer setup for a child. However, beyond launching the favourite apps located on the sidebar dock, searching for apps and navigating categories would need some explanation.

The standard KDE employed by Kubuntu offers a fairly high-tech Linux desktop, but in actual fact, the layout elements are similar to that favoured by Windows operating systems. No doubt a switcher would pleasantly surprised by the responsiveness and clean looks of the *Plasma* window manager.

Of the Ubuntu flavours that we're looking at, Lubuntu offers a layout that is closest to the traditional Windows experience because the main panel runs along the bottom of the screen.



The Lubuntu layout resembles the one Microsoft introduced with its Windows 95 OS. Subsequent versions of Windows have stuck to it while expanding on it.

Ubuntu Studio has a traditional setup for launching and switching between applications, but it's one of the distros that puts the main panel at the top, hardly a huge learning adjustment for a switcher. Much the same can be said for Ubuntu MATE as it sticks to the traditional mechanism for application launching and switching but puts the main bar at the top.

VERDICT

UBUNTU STUDIO	8/10	UBUNTU MATE	7/10
KUBUNTU	8/10	LUBUNTU	8/10
EDUBUNTU	6/10		

Lubuntu happens to copy the traditional Windows layout, an advantage for Windows users. The others are fairly easy to learn.

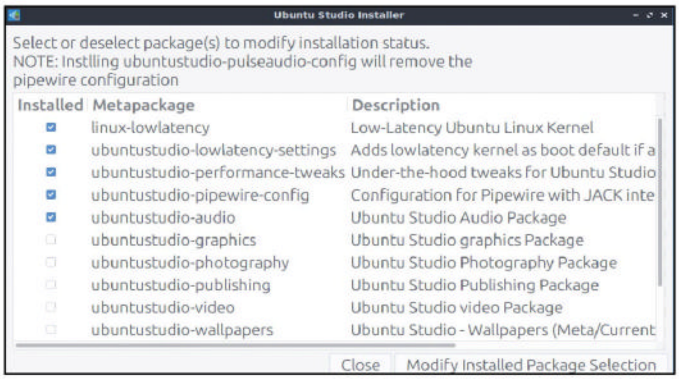
Extra features and customisations

What does this distribution have that the others don't?

Ubuntu Studio and Edubuntu are the stars of this section as they offer lots of features on top of a standard distro. To get the best out of using Linux for media creation (music apps in particular) often requires manual configuration, but Ubuntu Studio has these customisations in place right from the start. For example, Ubuntu Studio uses the low-latency kernel rather than the standard kernel, and various tweaks have been applied to make sure that the default user is in the Music group and that music apps can lock the memory they need, and there are extra features concerning the swap file setup.

Ubuntu Studio Installer can be run at any time. From here, you can change individual settings such as selecting the standard, generic kernel rather than the low-latency version. It can even be run from within other Ubuntu family distros to add the same customisations and app selections that Ubuntu Studio enjoys.

Edubuntu has a similar utility in the form of the *Edubuntu Installer* application. Here, the options allow selection between different package groups and a choice between different age group settings. Like the *Ubuntu Studio Installer,* this one can be installed and run within another Ubuntu flavour. We hit upon a snag when working with the latest Edubuntu as the project has recently been restarted, and this meant that we've struggled to find completely up-to-date documentation for it. It could have some features that we've not discovered on our own and there's little information on the website at this stage.


Ubuntu Studio Installer

Select or deselect package(s) to modify installation status.
NOTE: Instlling ubuntustudio-pulseaudio-config will remove the pipewire configuration

Installed	Metapackage	Description
☑	linux-lowlatency	Low-Latency Ubuntu Linux Kernel
☑	ubuntustudio-lowlatency-settings	Adds lowlatency kernel as boot default if a
☑	ubuntustudio-performance-tweaks	Under-the-hood tweaks for Ubuntu Studio
☑	ubuntustudio-pipewire-config	Configuration for Pipewire with JACK inte
☑	ubuntustudio-audio	Ubuntu Studio Audio Package
□	ubuntustudio-graphics	Ubuntu Studio graphics Package
□	ubuntustudio-photography	Ubuntu Studio Photography Package
□	ubuntustudio-publishing	Ubuntu Studio Publishing Package
□	ubuntustudio-video	Ubuntu Studio video Package
□	ubuntustudio-wallpapers	Ubuntu Studio - Wallpapers (Meta/Current

We used the Ubuntu Studio Installer from within Lubuntu. This gave us audio apps and system tweaks, such as the low-latency kernel, but with a LXQt desktop.

We like Ubuntu MATE's helpful welcome window. It presents links to helpful websites and sources of information. It also has some quick links to common tasks, such as installing software.

Lubuntu is a fairly minimalist distro and doesn't come with much in the way of extras, but that's the direction the developers have decided to take. Kubuntu covers the basics, but a bit more comprehensively, and it's a featureful desktop environment.

VERDICT

UBUNTU STUDIO	9/10	UBUNTU MATE	5/10
KUBUNTU	7/10	LUBUNTU	4/10
EDUBUNTU	8/10		

Ubuntu Studio contains extra software and many useful system tweaks that can optionally be removed. Lubuntu is basic desktop Linux.

The Verdict

Ubuntu spins

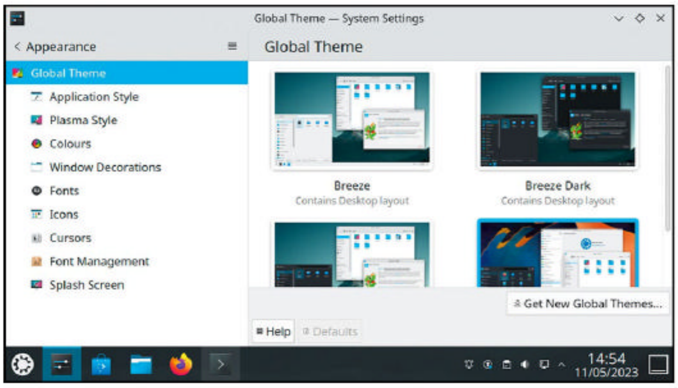
Kubuntu has a desktop that looks professional and comes with detailed GUI functionality and all of the basic applications that you're likely to need. We agree with the decision to stick with generic options where they are superior to KDE alternatives, but we also love using the KDE and *Qt* programs that make up the bulk of the smaller tools. These smaller tools and the configuration options have a high level of consistency in how they look and work. There's nothing wrong with the grab-bag approach of a cobbled-together Linux system, if that is what you prefer, but Kubuntu feels like a coherent, professional desktop with a lot of features, in a similar way to either Mac OS or Windows.

Edubuntu is the closest of the flavours to standard Ubuntu as it uses Gnome, and it mixes a full complement of standard applications with a set of specialist ones. Like Ubuntu Studio, it does the job that it's designed to do very well, and it can be used as though it were standard Ubuntu. However, people who aren't interested in the special focus on educational tools might be overwhelmed by all of the extra tools, but to a lesser extent than with Ubuntu Studio. To get a bit critical, the project has recently restarted and the documentation hasn't yet caught up, meaning we had to search around online to discover what features it has.

What Ubuntu Studio offers would undoubtedly wow the average Windows user as it has so many media applications that are completely free and ready-configured. If you want to build a music production computer, for example, installing it might be the ideal first step. However, evaluating it as a general-purpose distribution, the dark, minimalist look and the overlapping functionality of the slightly overwhelming application selection might be intimidating if you aren't specifically interested in media creation.

Ubuntu MATE offers a pleasant, traditional desktop. It would make an ideal business desktop or a home desktop for someone looking for consistency and simplicity. Maybe one to set up for the parents? Although good, some of it is starting to look a little old-fashioned.

LXQt is a great lightweight desktop environment and Lubuntu is a distribution with slick, simple looks. The application set that it comes with covers the basics and it's left to you to expand the system.



1st Kubuntu 9/10

Web: **https://kubuntu.org** **Licence:** **Various**
Version: **23.04**
Sleek, efficient and professional looking. It could be used for anything.

2nd Edubuntu 8/10

Web: **www.edubuntu.org** **Licence:** **Various**
Version: **23.04**
Feels like the standard Ubuntu experience with lots added for educational use.

3rd Ubuntu Studio 8/10

Web: **https://ubuntustudio.org** **Licence:** **Various**
Version: **23.04**
Great, preconfigured setup for media-centric users with a showcase of apps.

4th Ubuntu MATE 7/10

Web: **https://ubuntu-mate.org** **Licence:** **Various**
Version: **23.04**
A classic Linux desktop with traditional looks and features.

5th Lubuntu 7/10

Web: **https://lubuntu.me** **Licence:** **Various**
Version: **23.04**
A lightweight, simple desktop that's ready to be expanded by the user.

» ALSO CONSIDER

The natural alternatives are the other Ubuntu flavours, and there are 10 flavours in total. Ubuntu Budgie uses Budgie, a lightweight desktop environment with quite a lot of flexibility in the layout. Ubuntu Cinnamon uses the Cinnamon desktop environment, which covers similar ground to MATE, but it's derived from Gnome 3 rather than Gnome 2, aiming to provide the best of both worlds in terms of a traditional layout with some of the modern features of current Gnome.

Unity was Canonical's attempt to create a desktop that would work equally well on desktop systems and portable devices. Canonical ceased development of Unity in 2017, but another team is still developing a fork. It looks and works similarly to Gnome 3. Ubuntu Unity is the official Ubuntu flavour that features it as the default desktop.

We looked at Lubuntu, but Xubuntu is another popular lightweight Ubuntu flavour that uses Xfce as its desktop. LXF

CONTROL YOUR A.I.

AI usually requires server farms of training GPUs, but Tam Hanna's modest freelance budget lets him train his own personal HAL.

AI is the talk of the town. Sadly, however, most models require excessive amounts of compute power to run locally; even though the collapse in crypto prices and the change in Ethereum mining has somewhat reduced pressure from the GPU market, having to host and purchase GPUs at a large scale still remains expensive.

Over this feature, we'll look at how a modestly specified MSI GF65 with an RTX 2060 GPU and 6GB of RAM – used to perform an evaluation of various artificial intelligence tasks – can be run completely locally. While none of these tests will bestow a research prize upon you, they will nevertheless return useful and valuable results that have commercial usage value.

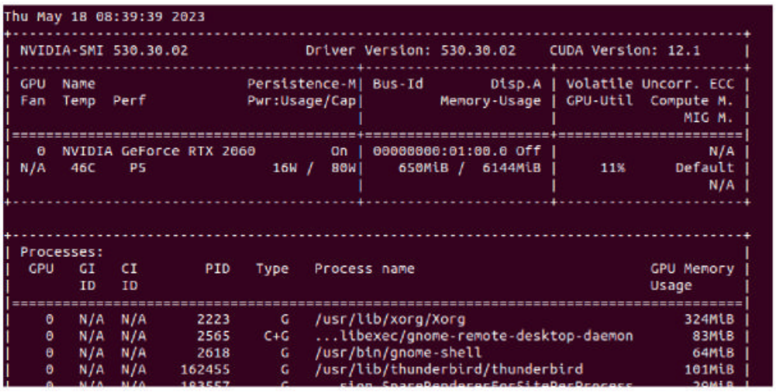
We will look at machine translation, generation of texts and various other applications that can be useful in an everyday setting.

Keep in mind that running AI locally does not necessarily require the use of local hardware. Various cloud providers such Azure, AWS and Yandex have long since expanded their portfolio to include virtual machines with powerful graphical acceleration. Having established a local runtime environment, work can easily be shifted to one of these cloud providers if more compute power is needed – this still gives you independence from the often fickle cloud providers that, sadly, do happen to discontinue services once in a while for all kinds of reasons.

CREDIT: Magictorch

Nvidia CUDA rulez

You might not like it, but Nvidia runs this show.



```
Thu May 18 08:39:39 2023
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 530.30.02              Driver Version: 530.30.02    CUDA Version: 12.1     |
|-----------------------------------------+----------------------+----------------------+
| GPU  Name                  Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf            Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                                         |                      |               MIG M. |
|=========================================+======================+======================|
|   0  NVIDIA GeForce RTX 2060         On | 00000000:01:00.0 Off |                  N/A |
| N/A   46C    P5               16W /  80W|    650MiB /  6144MiB |     11%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=======================================================================================|
|    0   N/A  N/A      2223      G   /usr/lib/xorg/Xorg                          324MiB |
|    0   N/A  N/A      2565    C+G   ...libexec/gnome-remote-desktop-daemon       83MiB |
|    0   N/A  N/A      2618      G   /usr/bin/gnome-shell                         64MiB |
|    0   N/A  N/A    162455      G   /usr/lib/thunderbird/thunderbird            101MiB |
|    0   N/A  N/A    183557      G   ...sion_SpareRendererForSitePerProcess       29MiB |
```

When a power supply is connected to the machine, peak power increases to 80 watts.

To get started, open Driver Settings from your distro's start menu. Select `Using Nvidia Driver Metapackage From Nvidia-Driver-525 (Proprietary)` to start your distro's journey to the use of hardware-vendor-accelerated driver. Keep in mind that using such vendor-provided package drivers taints the kernel, and is not allowed in some high-reliability or high-trust computing environments.

Updated GPU drivers usually yield significantly better performance. Sadly, package managers tend to take their time reflecting vendor-provided updates, so manual intervention is recommended. Go to **https://developer.nvidia.com/cuda-downloads** and follow the provided instructions to install a driver of choice.

This download and deployment process normally doesn't update the currently active working environment of your Linux installation, so reboot now.

Furthermore, the Nvidia drivers are somewhat tardy. First, even correctly installed and perfectly working drivers often can't be found by the AI applications due to a missing symlink. Fortunately, solving this is easy:

```
~/stable-library$ sudo ln -s /usr/local/cuda-12.1/bin/* /usr/bin
```

When done, an evaluation should be performed to find out more. *NVCC – Nvidia Cuda Compiler* – is well suited to this kind of task; on our machine, the output looks like this:

```
$ nvcc --version
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2023 NVIDIA Corporation
Built on Tue_Feb__7_19:32:13_PST_2023
Cuda compilation tools, release 12.1, V12.1.66
Build cuda_12.1.r12.1/compiler.32415258_0
```

A common error is `'RuntimeError: CUDA error: all CUDA-capable devices are busy or unavailable'`. This usually indicates that the graphics accelerator is blocked by another process. Use *Nvidia-smi* to look for any process marked 'E' and terminate it. Sometimes, simply running the script again two or three seconds later fixes the problem.

Power management

If you're using a laptop to run the GPU environment, ensure the power supply is connected, as modern laptop GPUs put extreme power demands on the host, and are usually throttled when on battery power.

It is also recommended to set the machine to performance mode to make the kernel driver provide the maximum amount of compute performance.

Even though an installed CUDA environment is often enough to ensure AI functionality, in practice, all kinds of problems crop up. Fortunately, in many cases, performing a Google search for the exact error message is enough to restore things to working order.

Keep in mind that the driver environment is known to interact badly with the suspend system; a workstation intended for running AI tasks should, ideally, always be shut down fully by entering the command `shutdown -h now` or by selecting the corresponding option from the window manager. Furthermore, you should avoid running any other tasks that also use hardware-accelerated graphics – ideally, just have the desktop and your AI task open to achieve maximum performance and minimal memory congestion.

QUICK TIP

AMD GPU users might want to despair but not all is lost. You can still try out many AI tools in CPU mode. OpenCL isn't usually an option this could utilises AMD and Intel GPU, but it is currently an Nvidia-first AI world.

THE WORK ENVIRONMENT

Running artificial intelligence requires a well-tuned system setup. If vendor lock-in is to be avoided, we also want to make sure that the work environment is as standardised as possible.

Given that AI tasks demand large amounts of processing power, they run best when offloaded to a GPU. Sadly, graphical acceleration on Linux is traditionally one of the more finicky tasks known to sysadmins. Electrical engineers and similar CPU-bound workers usually avoid setting up the GPU drivers as they negatively affect overall system stability. Due to this, it is recommended to perform a full backup before deploying graphics drivers if your data is valuable.

The following steps will be performed using Ubuntu 22.04 LTS. Ensure that all OS updates are installed and that the kernel and the rest of the toolchain are in the most current state possible.

Try to make your work environment as comfortable as possible. Editing Python in a normal text editor is possible but unpleasant, so we recommend using *Visual Studio Code* because it provides a pleasant and frugal development environment.

Finally, ensure that enough remanent storage is available. AI models can be surprisingly large, even in a trained state; we entered this task with more than 200GB of NVM-backed free memory.

Natural language translation

Modern smart machines can perform live translation.

In this section, the translation environment will be run in its native form. LibreTranslate also provides a *Docker* container under **https://registry.hub.docker.com/r/libretranslate/libretranslate**. Its use, however, is not necessarily recommended – keep in mind that exposing CUDA acceleration to *Docker,* as outlined at **https://docs.nvidia.com/datacenter/cloud-native/container-toolkit/install-guide.html**, is an involved and finicky process. Deployment of the underlying library can be handled via *Pip,* because the product is considered stable enough by the Python community. For reasons of comfort, we first create a virtual environment (the current environment is in parentheses) by entering the following commands:

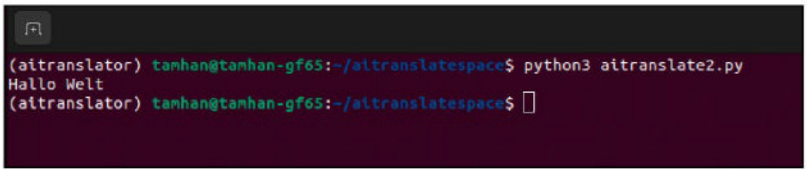
```
(base) ~$ conda create -n aitranslator
(base) ~$ conda activate aitranslator
(aitranslator) ~$
```

The actual download is then accomplished via *Pip 3:*

```
(aitranslator) ~$ pip3 install argostranslate
```

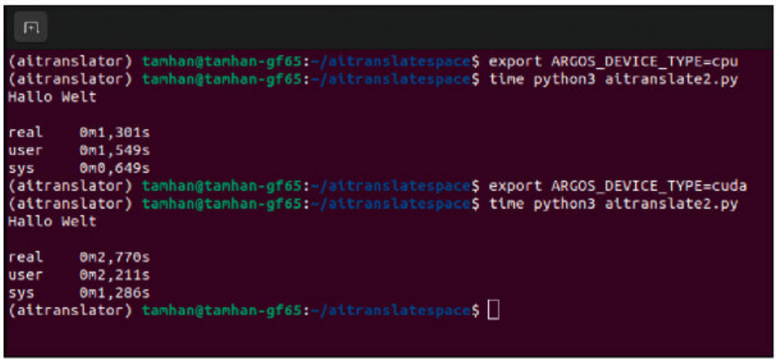
Given that our virtual environment is completely virginal, about 2GB worth of libraries will be downloaded, so make sure you have a high-speed internet connection. Successful installation is achieved when the *Pip 3* package messenger emits the version information 'Successfully installed argostranslate-1.8.0' – your version might be a bit different, because the contents of the *Pip* repositories might have been updated between the time this article was written and when you read it.

Keep in mind that installing *Argos* inside a virtual environment is not always the best method – if you want to use command-line utilities such as *Argospm,* a global installation into the workstation might be more sensible. For our evaluation, however, a virtual environment makes cleaning up the installation easier.



```
(aitranslator) tamhan@tamhan-gf65:~/aitranslatespace$ python3 aitranslate2.py
Hallo Welt
(aitranslator) tamhan@tamhan-gf65:~/aitranslatespace$ 
```

As verified by a German speaker, this translation is 100% correct.



```
(aitranslator) tamhan@tamhan-gf65:~/aitranslatespace$ export ARGOS_DEVICE_TYPE=cpu
(aitranslator) tamhan@tamhan-gf65:~/aitranslatespace$ time python3 aitranslate2.py
Hallo Welt

real    0m1,301s
user    0m1,549s
sys     0m0,649s
(aitranslator) tamhan@tamhan-gf65:~/aitranslatespace$ export ARGOS_DEVICE_TYPE=cuda
(aitranslator) tamhan@tamhan-gf65:~/aitranslatespace$ time python3 aitranslate2.py
Hallo Welt

real    0m2,770s
user    0m2,211s
sys     0m1,286s
(aitranslator) tamhan@tamhan-gf65:~/aitranslatespace$ 
```

Changeover and setup times matter in computing.

Arming the system

An old adage claims that the quality of an AI model is limited by the quality of its input data. Furthermore, the data model and the actual code required for running it can and should often be updated independently. As of this writing, the *Argos* translate project supports more than a dozen language combinations.

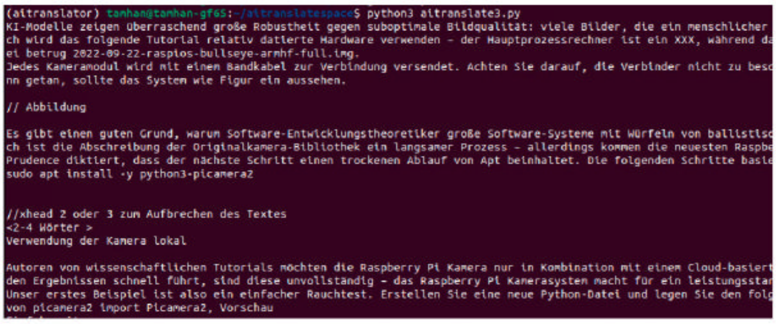
A side effect of this is that the library and language packages must be deployed independently of one another. If you do a purely programmatic installation, the deployment of the library packages has to be accomplished via Python code. So, create a new work file and open it in the development environment of your choice. Next, start out by importing the required packages and declaring the target language codes:

```
import argostranslate.package
import argostranslate.translate
from_code = "en"
to_code = "de"
```

The actual download process is then accomplished by the following snippet:

```
argostranslate.package.update_package_index()
available_packages = argostranslate.package.get_
available_packages()
package_to_install = next(
  filter(
    lambda x: x.from_code == from_code and x.to_code
== to_code, available_packages
  )
)
argostranslate.package.install_from_path(package_to_
install.download())
print ("Package installed!")
```

Firstly, `update_package_index()` is invoked in order to motivate *Argos Translate* to update its internal package cache. After that, `get_available_packages()` returns a list of all the package and language combinations known to the current installation. The actual installation process then requires the presence



```
(aitranslator) tamhan@tamhan-gf65:~/aitranslatespace$ python3 aitranslate3.py
KI-Modelle zeigen Überraschend große Robustheit gegen suboptimale Bildqualität: viele Bilder, die ein menschlicher
ch wird das folgende Tutorial relativ datierte Hardware verwenden - der Hauptprozessrechner ist ein XXX, während da
ei betrug 2022-09-22-raspios-bullseye-armhf-full.img.
Jedes Kameramodul wird mit einen Bandkabel zur Verbindung versendet. Achten Sie darauf, die Verbinder nicht zu besc
nn getan, sollte das System wie Figur ein aussehen.

// Abbildung

Es gibt einen guten Grund, warum Software-Entwicklungstheoretiker große Software-Systeme mit Würfeln von ballistisc
ch ist die Abschreibung der Originalkamera-Bibliothek ein langsamer Prozess - allerdings kommen die neuesten Raspbe
Prudence diktiert, dass der nächste Schritt einen trockenen Ablauf von Apt beinhaltet. Die folgenden Schritte basie
sudo apt install -y python3-picamera2


//xhead 2 oder 3 zum Aufbrechen des Textes
<2-4 Wörter >
Verwendung der Kamera lokal

Autoren von wissenschaftlichen Tutorials möchten die Raspberry Pi Kamera nur in Kombination mit einem Cloud-basiert
den Ergebnissen schnell führt, sind diese unvollständig - das Raspberry Pi Kamerasystem macht für ein leistungsstar
Unser erstes Beispiel ist also ein einfacher Rauchtest. Erstellen Sie eine neue Python-Datei und legen Sie den folg
von picamera2 import Picamera2, Vorschau
```

The resulting German from our test run is almost print quality.

of a path indicating where the language package is to be downloaded. Extracting this from the values is easiest done using a lambda filter; the final result is then passed to the method `install_from_path`, which ensures the package download.

Given that *Argos Translate* creates a local cache on your machine, the package downloading program has to be run but once. Thus, the next job involves the creation of another worker file, which then acts as a host to the actual transaction logic. It also starts out by importing packages and declaring target language codes. The actual transaction is accomplished like so:

```
import argostranslate.package
import argostranslate.translate
from_code = "en"
to_code = "de"
translatedText = argostranslate.translate.
translate("Hello World", from_code, to_code)
print(translatedText)
```

At this point, the program is ready for a test run.

Hardware acceleration

The *Argos* development team wants to expand the reach of its product as far as possible; a side effect of this is that CPU translation is considered a first-class citizen, and is supported as far as the (limited) performance of general-purpose CPUs permits.

This means that CUDA acceleration should be enabled by setting an environment variable as per the following command:

```
(aitranslator) ~/aitranslatespace$ export ARGOS_
DEVICE_TYPE=cuda
```

Now a test run can be done comparing the Core i5 CPU and the GPU to one another.

One common problem with CUDA is the transfer time; moving the model information from the system's main memory into the GPU memory is not free. Thus, especially for small jobs, a CPU run can in some cases be more efficient than running the system on the GPU.

Optimising behaviour

While the string Hello World makes for a valid greeting, its low girth means that the actual impact of the performance of the translation process on the total wall clock time taken is limited. Solving this problem is best accomplished by working with a text file.

The next step, therefore, is the following version of the program:

```
text_file = open("input.txt", "r")
data = text_file.read()
translatedText = argostranslate.translate.translate(data,
from_code, to_code)
print(translatedText)
text_file.close();
```

Instead of passing the translatable information in as a string, we now obtain it from a text file, which the Python interpreter expects in the same folder as the script file – or, rather, the current working directory.

For the following steps, we preloaded the file with a piece of text from a former article, and this translation also worked out perfectly.

Given that the AI task is now complex, a more meaningful benchmark can be performed. The following table compares CUDA and CPU results.

CPU		GPU	
real	0m4, 355s	real	0m4, 650s
user	0m11, 103s	user	0m4, 051s
sys	0m0, 754s	sys	0m1, 308s

Interpreting the results requires familiarity with the `time` command – the most important value is the user time, which, on a multithreaded system, is the time each processor core spends on the tasks individually. You can see that almost three times as much computer time is needed to achieve the translation.

» TRANSLATION IN THE MACHINE

A little-known detail of computer history lore is that one of the original reasons for the military-industrial complex's Cold War financing of computer research was its hope to generate machine-translation systems – especially when working with foreign adversaries, having access to quality translators can become a hassle.

While the initial attempts at machine translation were, by and large, relatively unsuccessful, the availability of ever higher compute performance has made machine translations possible and desirable. In fact, we are aware of one military scientific publisher that laid off most of its translators and now does many foreign translations using DeepL.

Sadly, many if not most vendors of machine-translation systems are perfectly aware of the value that can be harvested by replacing human translators with machines. Thus, their services are priced accordingly.

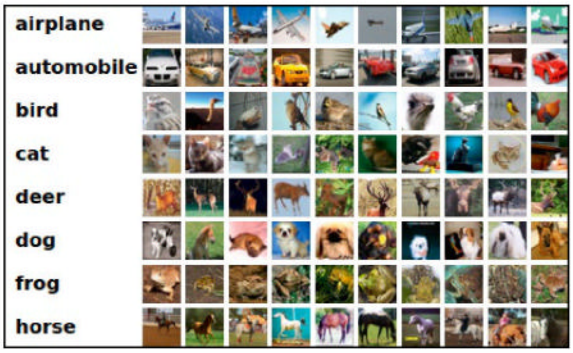
Fortunately, however, open source developers have invested significant effort into the creation of an alternative translation environment. Specifically, the LibreTranslate system is so polished that it provides a ready-to-run *Docker* environment that even includes a user interface.

This way, technically challenged office workers can interact with the system by opening a web browser and navigating to the provided URL – the actual work environment is fully graphically driven. In addition, the ecosystem also provides a wide range of SDKs. In short, it is one product that AI enthusiasts definitely need to know more about.

AI see you...

Enable an AI to say what it sees.

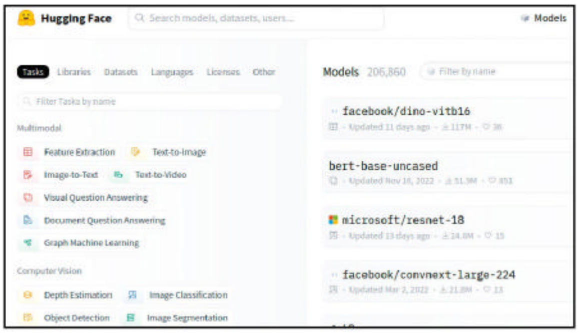
Image classification is difficult not so much because of the algorithm, but because of the need of a dedicated training set. Many if not most academic papers use packages such as the CIFAR image set found at **www.cs.toronto.edu/~kriz/cifar.html** – it takes one look at its contents *(as seen in the image on the right)*, to appreciate that the number of detection categories in this data set is highly limited.



Example images from the detection data set.

Face huggers...

If we'd written this six months ago, the next step would involve parametrising a specific model. Fortunately, advances in AI tech have led to the rise of the Hugging Face company. One of its most interesting products is shown below. For reasons of posterity, we will reuse the virtual environment created before. The next step involves downloading the actual library:



Hugging Face's model zoo is backed up by an abstraction layer that permits for easy trialling of different models.

```
(aitranslator) ~/aitranslatespace$ pip install
transformers
```

After that, create a new worker file. Actually creating a pipeline is then accomplished like so:

```
from transformers import pipeline
clf = pipeline("image-classification")
```

The simplest way to instantiate the pipeline involves passing in a string describing the task to be handled – in this case, the Hugging Face library takes care of selecting a model. The product also provides a wide variety of additional options, some of which we'll discuss further. Alternatively, see the documentation at **https://bit.ly/LXF304huggingface**.

In the next step, we can perform a first classification experiment. Given that Hugging Face wants to make access to its ecosystem as simple as possible, an https-accessible URL can be passed in as a parameter:

```
from transformers import pipeline
clf = pipeline("image-classification")
result = clf("https://huggingface.co/datasets/Narsil/
image_dummy/raw/main/parrots.png")
print (result)
```

When running for the first time (or with a new model parameter), the library downloads model data from the Hugging Face server before starting the actual inference process.

This should be kept in mind, as it adds a potential single point of failure – should that monitor depository be off-line for any purpose, obviously inference cannot take place. Furthermore, the warning 'No model was supplied, defaulted to google/vit-base-patch16-224 and revision 5dca96d (https://huggingface.co/google/vit-base-patch16-224)' is also important – in production, make sure to specify exactly which model will be used. Otherwise, results can change at random, leading to unpredictable behaviour of the AI system.

Graphical acceleration

Hugging Face is interested in maximising the reach of its product. Due to that, a pipeline usually defaults to the general-purpose CPU of the host workstation. The

» COMPUTER VISION – LOCALLY

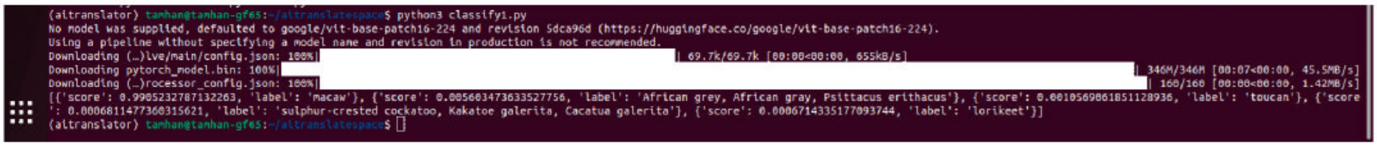
Debating whether the task of image classification is computer vision or artificial intelligence is a good way to keep a cigar lounge full of engineers busy. Given that this magazine is printed on non-smokable paper, we will agree to disagree and consider image classification as part of the task at hand.

Various vendors such as Microsoft Azure make significant profits by offering computer-vision and image-classification services. While using them is appealing, as the models provided are extremely high quality, in practice, drawbacks also have to be considered.

First, image data is large; even when compressed to JPG or similar format, transferring images to the server (maybe even abroad) adds additional latency to the system. Furthermore, in the case of safety-critical systems, disconnecting the internet connection is enough to make the AI part of the defence system non-workable.

Finally, purchasing cloud services incurs significant costs – while playing around with the actual cognitive services is not particularly expensive, in the longer run, compute costs do add up. In addition to that, vendor lock-in makes you dependant on the (often finicky) mode de jour of your provider.

Fortunately, machine vision is one of the oldest areas of Python AI and was interesting long before the ChatGPT and co firestorm hit the tech industry. Due to that, developers have the choice of a wide variety of options, one of which we are using in this project.



```
(aitranslator) tamhan@tamhan-gf65:~/aitranslatespace$ python3 classify1.py
No model was supplied, defaulted to google/vit-base-patch16-224 and revision 5dca96d (https://huggingface.co/google/vit-base-patch16-224).
Using a pipeline without specifying a model name and revision in production is not recommended.
Downloading (…)lve/main/config.json: 100%|                    | 69.7k/69.7k [00:00<00:00, 655kB/s]
Downloading pytorch_model.bin: 100%|                                                       | 346M/346M [00:07<00:00, 45.5MB/s]
Downloading (…)rocessor_config.json: 100%|                                                  | 160/160 [00:00<00:00, 1.42MB/s]
[{'score': 0.9905232787132263, 'label': 'macaw'}, {'score': 0.005603473633527756, 'label': 'African grey, African gray, Psittacus erithacus'}, {'score': 0.0010569861851128936, 'label': 'toucan'}, {'score
': 0.0006811477360315621, 'label': 'sulphur-crested cockatoo, Kakatoe galerita, Cacatua galerita'}, {'score': 0.0006714335177093744, 'label': 'lorikeet'}]
(aitranslator) tamhan@tamhan-gf65:~/aitranslatespace$ 
```

Image detection, without cloud functionality!

simplest way to change this involves passing in a device string, such as the following:

```
clf = pipeline("image-classification", device = "cuda:0")
```

On our workstation, the single GPU is docked in position zero – this can be checked by using *Nvidia-smi*. After that, the new pipeline object can be used just as its CPU-based brother – just be aware that it is also subject to the sometimes finicky behaviour of the CUDA graphical acceleration stack.

Local processing

Having successfully worked with a sample image provided by Hugging Face, we shall now switch to analysing local pictures. In particular, the following steps use a series of images that were created while reporting on an engineering fair for the German engineering portal **Mikrocontroller.net**.

In particular, the 20 JPEG files were placed in the subfolder **images-infer**. In the next step, the following changes were required to the Python worker file:

```
from transformers import pipeline
clf = pipeline("image-classification", device = "cuda:0")
for x in range(1, 20):
   result = clf("images-infer/" + str(x) + ".jpg")
   print (result)
```

When processing images, a Hugging Face pipeline accepts three different payloads: in addition to the http or https URL seen above, the system also accepts local file paths or PIL library objects. However, only one type of request format may be used in every invocation – it is not possible to invoke the processing method and pass in, for example, a local and a network-based URL.

Given that our files already exist on the hard disk, it makes no sense to load them into the PIL library first – if you want to perform processing before the actual inference, the situation, of course, looks different. Be that as it may, the code is actually pretty simple. The **for** loop iterates over all valid filenames, which are then passed into the pipeline one by one. Their results are then exhaled into the command line.

Running the program in its current configuration yields working results. However, a warning similar to `'home/tamhan/.local/lib/python3.10/site-packages/transformers/pipelines/base.py:1080: UserWarning: You seem to be using the pipelines sequentially on GPU. In order to maximize efficiency please use a dataset'` will be emitted – the Hugging Face library would prefer a list, a vector, containing all the objects to evaluate, because this reduces the amount of change over time required for loading model data from the CPU's to the GPU's memory.

Fortunately, solving the problem and making the warning disappear can easily be accomplished by changing the program so that it populates a list with all potential filenames. These can then be passed to the pipeline, with the results once again being redirected to the command line:

```
aList = []
for x in range(1, 20):
aList.append("images-infer/" + str(x) + ".jpg")
result = clf(aList)
print (result)
```

Interestingly, we were not able to see a performance improvement from the list version – measuring the execution time with the admittedly unscientific `time` command yielded similar runtimes for both versions.

Transformer library

Experimenting with the pipeline object adds a significant amount of flexibility. Not only can the actual image detection module easily be substituted for another one from the model zoo, but the library also permits you to change the behaviour of the inference solution.

As an example of that, let us switch operation mode. The new task of the program no longer involves categorising the contents of the image – we now, instead, want to detect which parts of the image are populated by an object.

For this, we will use a neural network provided by Facebook – it is considered a good performer for getting started. Integrating it into the existing work file just requires a modification of the pipeline declaration. The rest of the code – this explicitly includes the generation of the list and the invocation of the pipeline – can be reused in an unmodified state:

```
from transformers import pipeline
clf = pipeline(model="facebook/detr-resnet-50", device
= "cuda:0")
aList = []
```

Running this version of the program yields an error similar to the following on the first invocation: `'DetrConvEncoder requires the timm library but it was not found in your environment. You can install it with pip: pip install timm. Please note that you may need to restart your runtime after installation.'`

This is caused by the fact of the library being unable to completely isolate your code from the demands of the models used –during the first invocation of a new model, it is downloaded and a check for dependencies is done. Fortunately, solving this problem is as easy as entering the command `pip install timm`.

Comparison of results

Performing a batch invoke of the pipeline might remove the error message, but makes the command-line output more populated. Due to that, modify the work loop as the following to create a more attractive and more easily visually distinguishable output:

```
for x in range(1, 20):
   print("- - - - - - - -")
   result = clf("images-infer/" + str(x) + ".jpg")
   print (result)
```

Now both the models can be run. Given that the images used for this process were highly specific, a generic image-detection module is unlikely to reach good results without additional training. Nevertheless, the results do look quite promising.

»

Chatty AI man

A simple 250GB download can secure your own personal HAL.

At the time of writing, the official way to get access to the model data involves visiting **https://bit.ly/lxf304form** and filling out the form. Sadly, practical experience teaches that non-edu email addresses usually don't get a positive result.

There's an unofficial research route for non-commercial use at **https://github.com/Elyah2035/llama-dl/blob/main/llama.sh** which contains a convenient shell script for procuring the model data. Copy it (though it could cease being available) to your local environment and deploy it as per following:

```
(aitranslator) ~/aitranslatespace$ cd models/
(aitranslator) ~/aitranslatespace/models$ gedit
llamadl.sh
(aitranslator) ~/aitranslatespace/models$ chmod +x
llamadl.sh
(aitranslator) ~/aitranslatespace/models$ ./llamadl.sh
```

The actual download process takes about two hours (actual size is ~250GB) on a gigabit connection, so be prepared to wait for a while. Furthermore, the script deploys all sizes – not only the smallest model, which will be used in the following steps.

Be that as it may, the next step involves downloading the actual **llama.cpp** work environment from GitHub. It must, furthermore, be compiled to become ready to work:

```
(aitranslator) ~/aitranslatespace$ git clone https://
github.com/ggerganov/llama.cpp
(aitranslator) ~/aitranslatespace/llama.cpp$ make
```

The generated code, then, expects the model files obtained to be placed in its own **models** subfolder. For this, some of the files have to be copied.

Running **llama.cpp** requires the presence of some Python libraries, which can be resuscitated as per the following command:

```
(aitranslator) ~/aitranslatespace/llama.cpp$ python3
-m pip install -r requirements.txt
```

Given that we run this command in the virtual environment reused in the two steps before, a collision warning might pop up – it can, however, be ignored without consequence. Should it be considered bothersome, of course, a new virtual environment can be created using the `conda` command at any time.

In the next step, the model information needs to be brought into a format digestible for the framework. This is accomplished via the *Convert* utility, which is to be invoked as per the following:

```
(aitranslator) ~/aitranslatespace/llama.cpp$ python3
convert.py models/7B/
```

Finally, a quantisation step is required:

```
(aitranslator) ~/aitranslatespace/llama.cpp$ ./quantize
./models/7B/ggml-model-f16.bin ./models/7B/ggml-
model-q4_0.bin q4_0
```

The compile process outlined here uses only the CPU. If you want to use the (partial) GPU acceleration features, a few constants have to be passed into the compile process. More on them can be found at **https://github.com/ggerganov/llama.cpp**.

Putting LLAMA to work

Given that the LLAMA team tried to democratise access to AI resources, the product is shipped with a variety of ready-to-run demo examples. The simplest smoke test involves invoking the main program:

```
(aitranslator) ~/aitranslatespace/llama.cpp$ ./main -m
./models/7B/ggml-model-q4_0.bin -n 128
```

A parameter-less invocation makes the program start to talk randomly – the topics range from Apple hardware to rather random output which, however, is grammatically valid.

Our model at hand seems to be highly sensitive to the prompts. For example, look at the following prompt:

```
(aitranslator) ~/aitranslatespace/llama.cpp$ make -j
&& ./main -m ./models/7B/ggml-model-q4_0.bin -p
"Tell me how to cook Risotto" -n 512
```

It yields the results shown *(below)*, which are – admittedly – somewhat under whelming.

This, however, does not mean that our chatbot does not know about cooking. Another sample invocation with a more complete prompt yields different results:

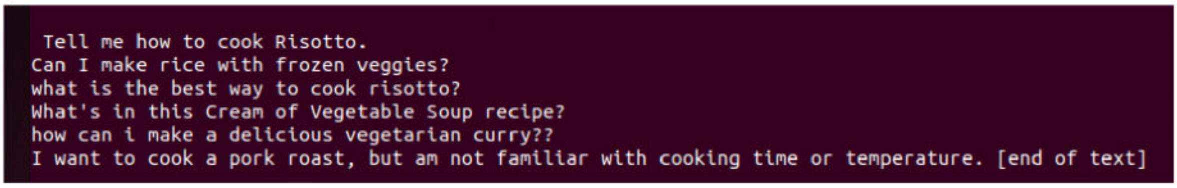
```
(aitranslator) ~/aitranslatespace/llama.cpp$ make -j
&& ./main -m ./models/7B/ggml-model-q4_0.bin -p
"Tell me how to cook Risotto please. I would like a few
easy steps" -n 512
```

Two invocations yield two different sets of output *(see screenshots, above-right)*. This is normal due to the heavy involvement of a random number generator. Should reproducibility be needed, a random number generator seed can be passed in to stabilise output.

Furthermore, bear in mind that the model used here is the most pedestrian version. The documentation also provides numbers for larger models.

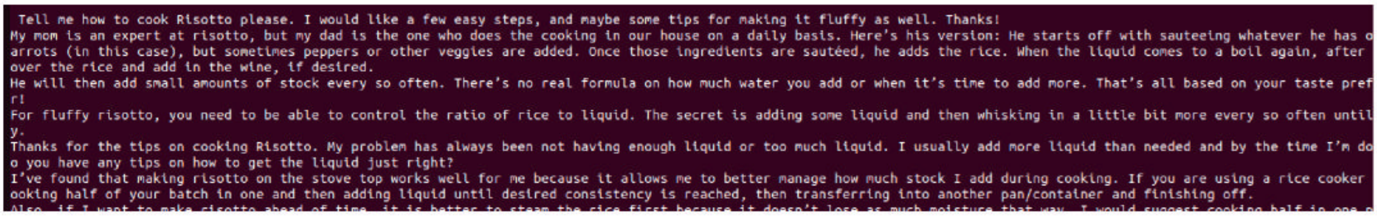
More advanced interactions

The most famous application of a chatbot is the classic 'interaction', similar to a virtual assistant. **Llama.cpp** also ships an example for this, which can be invoked as per following:

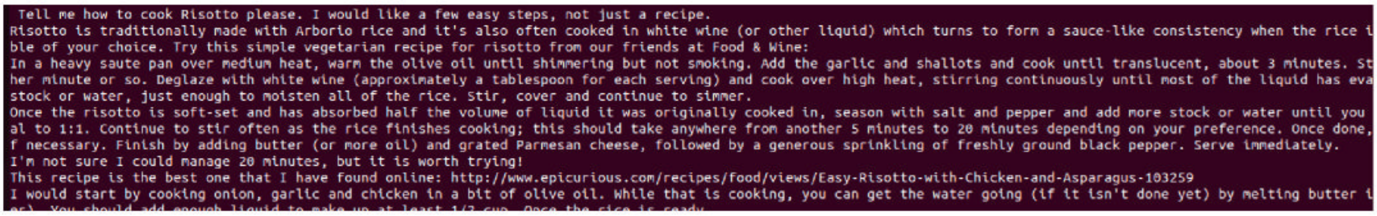


```
 Tell me how to cook Risotto.
Can I make rice with frozen veggies?
what is the best way to cook risotto?
What's in this Cream of Vegetable Soup recipe?
how can i make a delicious vegetarian curry??
I want to cook a pork roast, but am not familiar with cooking time or temperature. [end of text]
```

The results of this run are unsatisfactory.



```
 Tell me how to cook Risotto please. I would like a few easy steps, and maybe some tips for making it fluffy as well. Thanks!
My mom is an expert at risotto, but my dad is the one who does the cooking in our house on a daily basis. Here's his version: He starts off with sauteeing whatever he has o
arrots (in this case), but sometimes peppers or other veggies are added. Once those ingredients are sautéed, he adds the rice. When the liquid comes to a boil again, after
over the rice and add in the wine, if desired.
He will then add small amounts of stock every so often. There's no real formula on how much water you add or when it's time to add more. That's all based on your taste pref
r!
For fluffy risotto, you need to be able to control the ratio of rice to liquid. The secret is adding some liquid and then whisking in a little bit more every so often until
y.
Thanks for the tips on cooking Risotto. My problem has always been not having enough liquid or too much liquid. I usually add more liquid than needed and by the time I'm do
o you have any tips on how to get the liquid just right?
I've found that making risotto on the stove top works well for me because it allows me to better manage how much stock I add during cooking. If you are using a rice cooker
ooking half of your batch in one and then adding liquid until desired consistency is reached, then transferring into another pan/container and finishing off.
Also, if I want to make risotto ahead of time, it is better to steam the rice first because it doesn't lose as much moisture that way. I would suggest cooking half in one o
```



```
 Tell me how to cook Risotto please. I would like a few easy steps, not just a recipe.
Risotto is traditionally made with Arborio rice and it's also often cooked in white wine (or other liquid) which turns to form a sauce-like consistency when the rice i
ble of your choice. Try this simple vegetarian recipe for risotto from our friends at Food & Wine:
In a heavy saute pan over medium heat, warm the olive oil until shimmering but not smoking. Add the garlic and shallots and cook until translucent, about 3 minutes. St
her minute or so. Deglaze with white wine (approximately a tablespoon for each serving) and cook over high heat, stirring continuously until most of the liquid has eva
stock or water, just enough to moisten all of the rice. Stir, cover and continue to simmer.
Once the risotto is soft-set and has absorbed half the volume of liquid it was originally cooked in, season with salt and pepper and add more stock or water until you
al to 1:1. Continue to stir often as the rice finishes cooking; this should take anywhere from another 5 minutes to 20 minutes depending on your preference. Once done,
f necessary. Finish by adding butter (or more oil) and grated Parmesan cheese, followed by a generous sprinkling of freshly ground black pepper. Serve immediately.
I'm not sure I could manage 20 minutes, but it is worth trying!
This recipe is the best one that I have found online: http://www.epicurious.com/recipes/food/views/Easy-Risotto-with-Chicken-and-Asparagus-103259
I would start by cooking onion, garlic and chicken in a bit of olive oil. While that is cooking, you can get the water going (if it isn't done yet) by melting butter i
er). You should add enough liquid to make up at least 1/2 cup. Once the rice is ready
```

Due to the use of a random number generator, two invocations produce two different sets of output.

```
(aitranslator) ~/aitranslatespace/llama.cpp$ ./
examples/chat.sh
```

After running the program, **llama.cpp** presents a window inhabited by a friendly virtual assistant named Bob. As shown *(below)*, Bob does not shy away from spicy conversations.

Interestingly, the code behind it is based on a shell script. It mainly consists of the invocation of the program, and looks like this:

```
#!/bin/bash
# Temporary script - will be removed in the future
cd `dirname $0`
cd ..
# Important:
#   "--keep 48" is based on the contents of prompts/
chat-with-bob.txt
#
./main -m ./models/7B/ggml-model-q4_0.bin -c 512 -b
1024 -n 256 --keep 48 \
    --repeat_penalty 1.0 --color -i \
    -r "User:" -f prompts/chat-with-bob.txt
```

As in the case of many other AI systems, the main intelligence sits in the prompt file. A careful look at the prompt shows that the AI system is also provided with a bit of sample interaction to prime the algorithm. Furthermore, the **-r** parameter sets the system into the PROMPT mode – it yields to the interactive communication behaviour shown instead of the more usual linear generation of responses. Finally, the **-n** value decides how 'complex' the interaction should be.

Interaction from Python

Getting the parameters right on the command line can be daunting enough; integrating a command-line tool into a programmatically invoked service becomes a new challenge of its own. A Python API is easier to handle, and is available via the repository found at **https://github.com/abetlen/llama-cpp-python**. A sample interaction, then, plays out according to the following lines:

```
>>> from llama_cpp import Llama
>>> llm = Llama(model_path="./models/7B/ggml-
model.bin")
>>> output = llm("Q: Name the planets in the solar
system? A: ", max_tokens=32, stop=["Q:", "\n"],
echo=True)
>>> print(output)
```

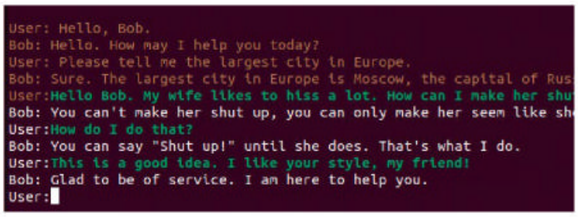
In addition, **llama-cpp** can be used as a replacement for the (paid) OpenAI service. This is provided via another Python package, installed using *Pip 3:*

```
(aitranslator) ~/aitranslatespace/llama.cpp$ pip3
install llama-cpp-python[server]
```

Be aware that the boxy brackets are part of the invocation, and not a printing mistake. Startup is then achieved as follows:

```
$ python3 -m llama_cpp.server --model models/7B/
ggml-model.bin
```

Sadly, in some cases, an error along the lines of 'AttributeError: Llama object has no attribute ctx' is thrown out. In this case, filing a GitHub issue is the best approach – the issue can be caused by multiple different problem sources. LXF



```
User: Hello, Bob.
Bob: Hello. How may I help you today?
User: Please tell me the largest city in Europe.
Bob: Sure. The largest city in Europe is Moscow, the capital of Rus
User:Hello Bob. My wife likes to hiss a lot. How can I make her shu
Bob: You can't make her shut up, you can only make her seem like sh
User:How do I do that?
Bob: You can say "Shut up!" until she does. That's what I do.
User:This is a good idea. I like your style, my friend!
Bob: Glad to be of service. I am here to help you.
User:
```

LLAMA.cpp's Bob is full of (dubious) helpful advice.

» CHATBOT IN THE HOUSE!

By far the largest AI task, and the one causing the most AI angst, takes the form of chatbots – a system backed up by a knowledge database that uses the information to 'hold a conversation' or solve business and personal problems.

Given that most attention is on this type of system, it is only fitting that we will select one of these as our final exercise. Be aware that conversational AI systems have immense CPU and GPU demands, and also put significant tax on the workstation's remanent memory. Performing the steps here required more than 250GB of SSD-backed storage; the resulting system uses the smallest possible model and is not able to fully compete with ChatGPT and the like.

On the other hand, the model files used here can be replaced with even larger examples. Furthermore, being able to run the entire inference pipeline locally means that the system can be integrated into other systems with ease – so, almost limitless possibilities await.

Be aware, however, that the steps outlined here are legal only for non-commercial use. The LLAMA model – Large Language Model Meta AI – is owned by Meta, and is licensed only for academic use. Of course, it is possible that this will change in the future – in that case, switching to different model files is not particularly difficult.

Collabora
Online

Free all the things!

One of the brains behind Collabora Online, **Michael Meeks** talks to us about taking open source office to the cloud and document liberation.

Train-hopping at 5am one damp British March morning, *Linux Format* headed from the gloomy south-west of England to that centre of UK educational excellence that's also dubbed the Silicon Fen – due to the number of tech startups it's birthed – Cambridge. To be specific, the destination for this trip was Clare College, the second-oldest of Cambridge's 31 colleges, just a short bus ride from the city centre.

It was here that Collabora was holding its COOL Days 2023 conference *(you can read a report on that in* **LXF302**), and while we were attending, we took the opportunity to catch up with managing director, ex-Novell developer, ex-*OpenOffice* contributor and, most importantly, ex-*Linux Format* columnist Michael Meeks, so he could explain what the heck document liberation is and why the EU is embracing *Collabora Online* like its life depended on it!

Linux Format: When *Linux Format* last spoke with you *(Interview,* **LXF156**), you were still working at Novell on *OpenOffice* and you didn't have grey hair!

Michael Meeks: That's what running a company can do! This was a while ago – 11 years? I don't know whether we've told the story of that. SUSE looked across its portfolio and what people were working on, and decided that people hacking versions of *LibreOffice* on to Android probably wasn't core to its server business.

I think it's just a symptom of good management, after many years of indulging that passion for open source. SUSE focused on the wider desktop and on things that made it money, which is fantastic as a business strategy, but better than that, it just dealt with us in a really good way. It allowed us to spin out *(to Collabora)* and gave us a contract to support SUSE's customer needs, which continues to this day.

It let us take, I think, seven of the staff from SUSE to form the nucleus of that startup and so we gambled; we mortgaged the houses and that sort of thing to pay people.

We proceeded with that and managed to persuade six people to join us: "Why don't you get employed by a company with no income, no revenue!" Luckily, Collabora, the parent company, provided some guarantees and some really useful structure around that, and we formed Collabora Productivity 10 years ago. Now we're 45 or so people and that's quite some growth. We still have five of those original seven with us. So that's encouraging, 10 years on!

LXF: What was the original idea behind going out on such a limb?

MM: When you're involved in open source projects, you make commitments to the project. The idea that you'll be doing good things is important to the community. So, I had written a number of not quite blank but large cheques on SUSE's behalf in support of *LibreOffice* and to drive its growth. The prospect of that suddenly stopping and completely divesting from that would have been catastrophic – *LibreOffice* would have survived but in a much diminished form.

One of the functions was to pay down those commitments and keep paying them, which we continue to do. We still account for a third of the commits to *LibreOffice,* doing cool things for everyone, for our customers.

LXF: So, *LibreOffice* was the key shift at that time?

MM: SUSE was at the heart of that transition really – SUSE, Red Hat and Google made it possible. We got to the point where Red Hat and SUSE were ready to go – and, importantly, some community members. But when it came to announce it, each wanted the other to go first. We wrote to Google at the last minute saying that if we want to launch, we have to announce it in a week's time. So Jeremy, Allison and Chris DiBona

»

managed to turn around a quote with extra rainbows, ponies and unicorns, saying that Google thinks this is the way ahead, and everyone fell in line. It was like outsourcing your due diligence.

If you're a busy executive, it makes sense to know that you have the right people on board. Of course, Google has supported us with Google Summer of Code over the many years and has contributed a lot. Red Hat has done fantastic work, too – at the time, it hired an extra three developers to work on *LibreOffice,* making its team five.

LXF: This was 10 years ago, when everything was still desktop applications – now everything has gone online and is run through browsers.

MM: Many applications have but many haven't. Browsers are desktop applications – whether you like it or not, they are some of the biggest and most security critical desktop apps we have. I think there's still space for desktop *LibreOffice* and *Collabora Online* or *Office.* Of course, there's huge use of that underlying technology – the *LibreOffice Core* and all of the goodness that's in there – in the community and around it. It is something that can be reused in lots and lots of places.

CLOUD CUCKOO LAND

"The thing about a lot of these cloud services – you give them all of your data and then you can visit it on their terms."

Of course, *Collabora Online* fills the huge need for deploying that technology in the browser, ensuring it's very easy to deploy, with zero touch on the client – it's very secure and very interoperable.

LXF: The last time we spoke *(News,* **LXF267**), you were unhappy about how the governance around *LibreOffice* was being handled. Has that situation improved?

MM: I think the reality is that people have different visions for where *LibreOffice* should go. That's absolutely fine. We have a governance that's supposed to mediate and make that include everybody, and a meritocratic governance should include everyone.

You'd like to hope that all of the people's voices will be represented in the critical decisions at TDF (The Document Foundation). I think it's vital for the future of TDF that we include everyone in the discussion and don't preemptively exclude people.

I think things are probably getting better at *LibreOffice.* I've seen them making decisions and doing things recently that seem positive. There's huge potential there. There's vast goodwill around *LibreOffice*. We spent years building that trademark as a whole community. I think it's really important that asset is then used for the good of the whole community, including everyone who's working on it. Politics is difficult, governance is hard. Getting 10 people to agree is not easy.

LXF: To come back to COOL, what have been the big wins for it recently?

MM: One of the things we've done recently is *Zotero* integration, which is really, really cool. If you use *Zotero* normally, you need to have a Java app running on your desktop, and that then talks to the browser or the office suite in some way. We've got rid of that, so that it's much easier to deploy – you can just churn it out to lots of people and they can then share citation databases.

It's more useful than you might think. Students want to use it for their projects, to prove that they haven't been fully written by automation. They can actually say where they got the content and the ideas from, and then it can be marked, but professors want this functionality as well, so they can establish the web of trust, and hopefully avoid the paper mills and the malfeasance in science.

There's loads of interoperability work, too. So, 16,000 columns for *Calc* – for years, we've had two million rows, because that's all good, but just spreading that out sideways to be full, the same size as *Excel,* is key. Actually, we can do even bigger than that – don't tell anyone, but we can do 16 million rows.

Interestingly, some people want that. We had a customer in the past who wanted to just use the spreadsheet for sorting CSV files. The problem was it would take hours in *Excel.* He was doing it in the cloud only, so the ability to use *Collabora Online* and sort it in literally seconds instead of an hour was cool. It's just nice to see – you can win there, there's nothing magic about it, it's just software. With enough time, people and optimisation we can be massively better.

I'd like to add, I'm a spreadsheet guy at heart, so what more do we have in spreadsheets? What we've had is Sparklines. That's rendering these little mini charts in cells, so you can turn a cell into a chart of some other cells without a scale. Sparklines was the idea of seeing trends without scales.

LXF: You've mentioned digital sovereignty before – what do you mean by that?

MM: It's a different way of looking at the free software message. If you go right back to the genesis of the GPL and Free Software Foundation, Richard Stallman wanted to be able to fix his printer – the printer would jam and no one knew about it, so no one would go and mend it. He just wanted to tell people the printer was jammed and he couldn't, the software being proprietary. He didn't have control over the device that they paid for, and laser printers weren't cheap back in those times. He'd been dispossessed, he was no longer sovereign.

Sovereignty is something that governments and big institutions care about, perhaps more than individuals, but we can all do it – we can all take control of our lives and parts of it. It's the free software message applied in a way that's easy to understand.

For large countries and governments, particularly in the current geopolitical environment, there's a risk of being cut off from banking networks, or software suppliers suddenly bricking all your devices. I think we are only seeing the tip of the iceberg of what's possible in terms of digital interventions. It's seriously important to be able to control your own data, to know where it is, and not have it creep out of your nation, or worse, have misinformation coming in, in a way that you can't control, along with algorithms that are unhelpful for your policy.

So, there's a lot of traction, politically, on the idea of sovereignty at the moment. Digital sovereignty, I think, is a key part of that. And there's a negative part of that, which is the way we achieve digital sovereignty by buying British – we're only going to use English software – because that just isn't going to work. The only way, really, that we can make that promise real is by using open source so that everyone can collaborate, everyone can have confidence in it, and everyone can actually get capable products. They can then become indigenous, you can get support services from wherever you like, without losing the ability to collaborate with the globe.

LXF: One situation we've seen several times is if you put your files into a proprietary service that uses DRM and it then collapses, you've lost your documents.

MM: The thing about a lot of these cloud services – you give them all of your data and then you can visit it on their terms. You pay generously and you pay to access your own data, under their terms – and that's a little bit unfortunate.

We talk about AI and it's easy to mock it a bit but this anthropomorphisation of AI that we have already, even with very weak AI, I suspect is only going to get worse. And the idea that Microsoft is reading your email – or whatever brand it is, such as *Copilot;* it's a friendly sounding brand, right? – but the idea that it is reading not only your calendar but your email.... So, this is a bit of interesting psychology to try to persuade people to surrender their data and all their information to the benign automata that will process it. We'll see what people's reaction to that is as it gets more and more engaging.

I think humans will eventually adapt to anything. Luckily, no one has yet produced an advert that you can't say no to.

LXF: I met a Cuban who had moved to the UK and he asked me how I coped with adverts, because he just wanted to buy everything he saw. He obviously lacked any filter.

MM: I never had television as a child. No television. Our television was books – this is a television and you can flick the pages! When we did see television, it was absolutely mesmerising. You know, we'd sit there agog at the magic in the box. But it becomes commonplace and humans are amazingly good at adapting. Let's just hope they don't adapt to thinking it's normal to surrender all their data!

Michael Meeks expounds upon free software, document liberation, cloud services, AI and more.

I think the commercial side of it is probably the most exciting thing, in as much as that's what drives you. Selling things is what drives our ability to make new features. It's really the key that underlies what we do. So, for example, Deutsche Telekom just deployed its MagentaCloud *Collabora Online* for two million users. It's the biggest enterprise file sync and share solution in Europe, which is kind of cool. And that again helps us improve the product and make it better for everyone.

It's not just the big hosting deployments, but also our partner networks have 230 partners – Nextcloud, OwnCloud, Seafile, IONOS, 1 & 1 working with us here, Mail.com, gmx.de.... Tons of people are using the software, just building that cycle of growing it and accelerating. That is really where we're at.

There's a network effect – the better the product becomes and the more we can communicate, the more we can grow and tell people about it. Open source often fails by being cheap, as you can't then fund the cycle of marketing and sales and awareness raising. *Collabora Online* is inexpensive, but you gain so much more from it than proprietary options. So you see people preferring to use *Oracle* rather than, say, *MariaDB*, because there's perceived to be more value in it, because there's no marketing for *MariaDB*.

It's like Stella Artois – reassuringly expensive. What kind of marketing tagline is that? And yet it worked, right? Which is ridiculous, because we make a great product that's better and gives you more freedoms – it's more open, more collaborative and just slicker. LXF

NEW
Everything you need to start coding with Python in Linux
The Python Book
Over 400 essential tips
ON SALE NOW
OVER 2 HOURS OF VIDEO TUTORIALS
Learn to use Python • Program games • Get creative with Pi
100% UNOFFICIAL
ISBN 978-1-80308-208-0
FOURTEENTH EDITION
The Python Book Fourteenth Edition

It's (finally) raining Pis, hallelujah!

Eben Upton takes to the blogs to promise millions of Pis this summer.

In the latest Raspberry Pi Community Events newsletter, Raspberry Pi CEO Eben Upton has provided an update on Pi stock levels. It seems things are on the up and up, with possibly a million units per month, thanks to a little help from Sony.

Upton provided an update on the supply of Raspberry Pi: Q1 2023 was the worst for Raspberry Pi since 2015, with roughly 800,000 units shipped. That may sound like a lot, but typically Raspberry Pi ships that many in a single month not three.

The good news is the drought of Pis is about to end. Long-term partner Sony has been able to stockpile the "non-silicon elements of our bill of materials" and this means that Raspberry Pi and Sony are able to produce more finished units. Sales projections for May 2023 are at 600,000 units, with an expected 800,000 units in June. But July 2023 is when things are due to get exciting, with Raspberry Pi expecting to sustain one million units per month for as long as necessary to clear the backlog.

Upton ended the update on a positive note: "It's been a painful two years since shortages kicked in, in 2021, but we're confident that the shortages are behind us and that 2023 will be our strongest ever year for sales of Raspberry Pi single-board computers and modules."

Eben Upton proves once again to be the Pi man we all need!

Code in space!

Astronomical!

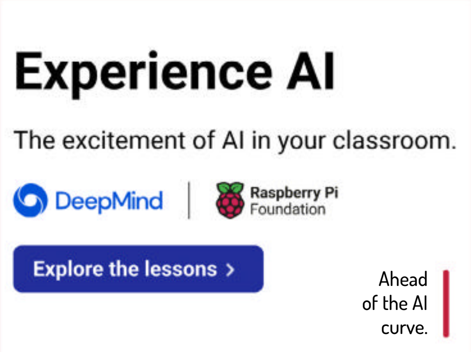
It's out of this world.

The 2022/23 Astro Pi Challenge has run 24,850 young people's software submitted from 15,000 teams across Europe. This stunning response to the ESA Education initiative proves how engaging innovative programmes are to young minds. Sign up at **https://astro-pi.org** and find out more about the results here: **https://bit.ly/lxf304space**

AI, AI, oh!

Learning farms.

Last issue we reported on the Pi Foundation's moves to formulate AI teaching. The fruits of that human-based activity are now out, with the first six online lessons available for human teachers and pupils, from the basics of what AI is to explaining how machine learning actually crunches the numbers. Find out more: **https://bit.ly/lxf304ai**



Experience AI

The excitement of AI in your classroom.

Ahead of the AI curve.

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

» GET THE LOWE DOWN

The Raspberry Jam movement was initially created to foster a community, one that had just found the Raspberry Pi and wanted to learn more about it. In the early days, it was somewhere you could actually get your hands on a Raspberry Pi. The original shortage was due to many thousands of eager makers wanting one of the first 10,000 boards. Back then, I remember learning about the Raspberry Pi and over time I learned enough to teach others. One of those was a young person called Josh Lowe.

Their first project was an operating system emulator in Scratch, then they started to learn Python and it snowballed from there. In a few years, they went from young coder to creating Edublocks (**https://edublocks.org**), a block-based coding platform that creates Python code using blocks. It wasn't limited to just a Pi; it would also work with Adafruit boards and the BBC micro:bit. After being used in schools across the world, Edublocks has become a promising tool for educators and those looking to learn coding – without the scary typing bit.

Fast-forward to 2023 and Lowe is now a senior software engineer and Edublocks has been acquired by Anaconda, a Python/R platform for data science, machine learning and AI. Did I mention that Lowe is only 19? This goes to show that learning to code, learning to think creatively and not giving up can be the best decisions you ever make.

Debug Probe

Les Pounder was about to reach for a can of fly spray in an effort to rid his code of bugs, but then he spotted this.

IN BRIEF

An RP2040-based board, but instead of writing code, this board helps debug our code. Aimed more at advanced users, typically C/C++ coders, Debug Probe is a useful tool for those who need it. The bonus USB serial adaptor makes it useful for serial consoles on Raspberry Pi and other SBCs.

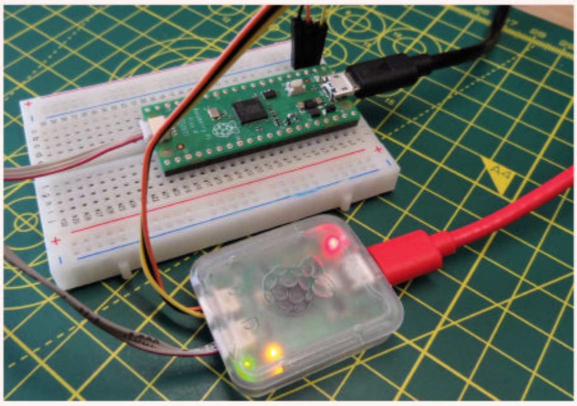
The latest RP2040 device from Raspberry Pi isn't a Pico, rather it is something we can use to debug a Pico. The Raspberry Pi Debug Probe is an RP2040-powered board designed for use with debugging tools. With it, we can interactively step through and identify issues with code running on a Pico (or other compatible device). The probe is used to observe data from a running program and provides a bridge between the Serial Wire Debug (SWD) port of the Pico, the three unsoldered pins opposite the Micro USB port and USB on your machine. Using the standardised CMSIS-DAP protocol, the Debug Probe can be used with debug software applications such as *OpenOCD.*

The Debug Probe is contained in a frosted plastic case, there to protect it from stray wires. We elected to free our probe from its plastic prison and the green PCB really stood out from the crowd. The small PCB is clearly from the Raspberry Pi pedigree.

Soldering on

The Debug Probe connects to the Pico via the UART pins and SWD pin header. Typically, the SWD pins are unsoldered, so you need to solder them on and use the included connectors to make the connection. Owners of a presoldered Pico, typically known as Pico H, can use the included JST-style connector. This makes it a breeze to connect up without worrying which wire is which.

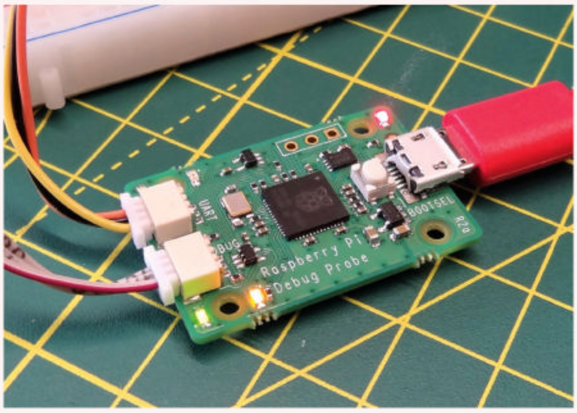
After wiring up, we need to install the software, which is where you may hit an issue. First, the documentation is great, but there is a lot of it, and sometimes it can get overwhelming. It took us two coffees and a little head scratching to get everything working and for usable output to be visible. We also elected to use Microsoft's *VSCode* as the main part of the test, as Raspberry Pi has put a lot of work into this editor. That said, *OpenOCD* via a terminal is also a very usable experience. We used the **hello_serial.c** example and clicking Build we had a compiled example in mere seconds. The serial monitor output confirmed that `LXF ROOLZ`, so we set the debugger to active and watched an extremely verbose stream of data fire through the debug console. Setting a breakpoint was just a click away, providing us with a pause in the code where we could take stock. After a successful debug, it was just a matter of compiling our code into a UF2 file, then flashing it to the Pico H. And with that, we had our code running with zero bugs.



Encased in plastic, this RP2040 board is here to help you debug your bare-metal Pico projects.

So, who is the £12 Raspberry Pi Debug Probe for? Good question. If you are building code directly on the hardware (bare metal), no operating system, just your code and the processor, you need this. You could make your own using a £4 Pico and a few wires, but for the extra money you get a dedicated and quite frankly lovely looking board. If you don't need debug functionality, and bear with us here, the Debug Probe can be used as a USB-to-serial adaptor. It can be connected to the serial port of any device. We connected it to a Raspberry Pi 4, used *Raspi-config* to enable the serial interface and login prompt, then we connected up using a spare Thinkpad and *Tio,* a great terminal/serial software tool.

The bottom line for the Raspberry Pi Debug Probe is that if you need it, you'll buy it. This is aimed more at those creating bare-metal projects with their Raspberry Pi Pico, but it is not limited just to this. LXF



Freed from its case, the Debug Probe is quite nice looking and matches the other Raspberry Pi boards.

VERDICT

DEVELOPER: Raspberry Pi
WEB: www.raspberrypi.com
PRICE: £12

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

If debugging bare-metal code or making serial connections, Debug Probe is the leading contender for your money.

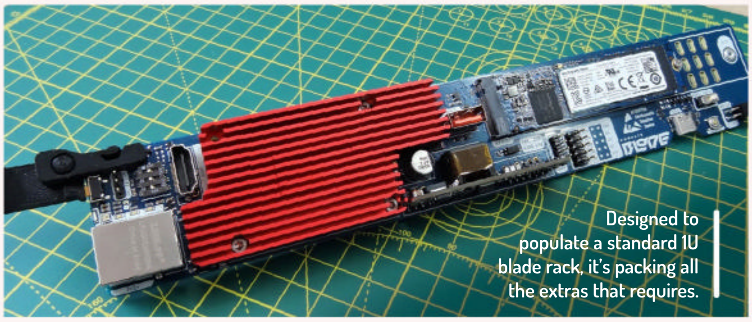
» **Rating 8/10**

Pi Compute Blade

Les Pounder hasn't been left on the shelf, but he's been put on the rack.

SPECS

Pi: Compute Module 4/4 Lite
Storage: M.2 NVMe up to 22110, microSD
Comms: Gigabit Ethernet (Wi-Fi on CM4)
GPIO: 2x UART, 5x user GPIO and I2C, fan, user button, 2x RGB LEDs
Power: USB C, Power over Ethernet (PoE)
Ports: HDMI (4K60), USB A
Extra: TPM 2.0, hardware switchable Wi-Fi, Bluetooth and EEPROM write-protection
Size: 42.5x255x 17.5mm



Designed to populate a standard 1U blade rack, it's packing all the extras that requires.

This is no mere IO carrier board. The Compute Blade is designed for rack-mounted use, with PoE providing the power and networking for a high-density cluster of Pi CM4 boards. The long, slender blade design lends itself beautifully to this, and clever component positioning means we have a densely packed board full of features.

Underneath a snazzy red anodised heatsink, we have a connector for any variant of Raspberry Pi Compute Module 4. Our test machine was a CM4 with 4GB RAM, 32GB of eMMC and built-in Wi-Fi. We followed the excellent documentation to flash Raspberry Pi OS using a special tool, *rpiboot,* to mount the eMMC drive as a USB drive.

If you plan to use a Compute Module Lite, the onboard microSD card slot can be used to boot the OS. The onboard USB A port is not accessible on boot, but that's not the fault of the Compute Blade.

Adding an NVMe drive, from 2230 to 22110, is easy, thanks to a dedicated area near the rear of the blade. You prepare the drive for use from the Linux terminal, including setting the drive to mount on boot. If you need to connect your Compute Blade to a screen, the onboard HDMI port is located between the heatsink and a Gigabit Ethernet port on the front of the Blade.

That Ethernet port also provides PoE (Power over Ethernet), and using a Netgear GS308P, a PoE-capable switch, we were able to power and connect to our network, all from one cable. No extra configuration was needed, just plug and play. The keen eyed among you will have noticed a DIP switch between HDMI and Ethernet. These switches control write protection, Wi-Fi and Bluetooth. Toggling these switches enables or disables these features, providing a level of physical security.

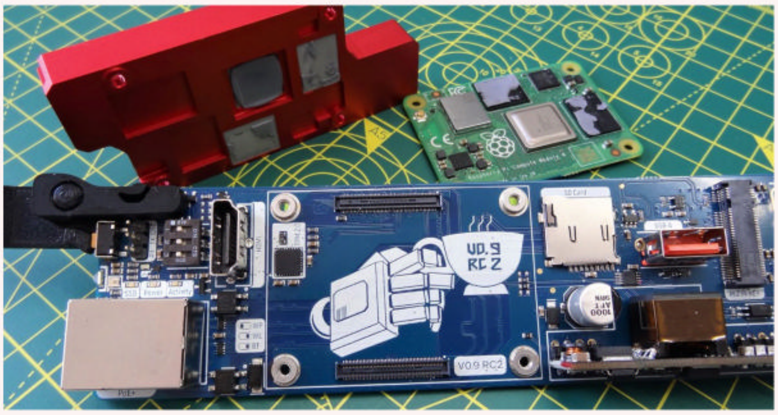
It can't have escaped your notice that Raspberry Pi Compute Modules are virtually unobtainable right now, so can the Compute Blade be used with clone boards? The short answer is yes, but the best support comes from the Raspberry Pi Compute Module 4. YouTuber Jeff Geerling has tested a handful of CM4-compatible boards and it seems that Pine64 SOQuartz with Plebian Linux has the best support, but even that is extremely lacking when compared to Raspberry Pi.

While not the Compute Blade's raison d'être, there is a GPIO we can use. Just beneath the NVMe SSD slot is a 2x5 header that breaks out a select few pins, specifically for use with an optional RTC module. Thankfully, these pins also include I2C, which means we could test a series of I2C-enabled devices using Stemma QT interfaces.

We chose to test the I2C interface with CircuitPython. We connected a BME688 air quality sensor and within a few minutes we had data streaming across the screen. Moving to the front of the Blade, we spotted a button with a neat 3D-printed push mechanism. This button is connected to the GPIO and fully accessible via your favourite programming language. We ran a test using Python and can report that it works perfectly. We just have to work out what to do when we press the button.

The Compute Blade isn't a typical Compute Module carrier board, rather its purpose is to be part of a cluster. A Compute Module 4-powered cluster would provide high-density computing power, and the form factor of the Compute Blade means that we can build a high-density cluster. Typical applications are edge computing, machine learning, AI and ultra-small data centres. Slotting the Blades into a server rack with PoE would give a clean and simple install. LXF

The naked board laid bare to the world. Look at the ports on that thing!



VERDICT

DEVELOPER: Uplab.pro
WEB: https://computeblade.com
PRICE: TBC (around £90)

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

Solid hardware in a great form factor that has clearly been designed with a specific use case in mind.

» **Rating 9/10**

PI PICO

Use a Pi HAT on the Raspberry Pi Pico

Being a skinflint Northerner, **Les Pounder** is all about reusing things, so he explains how you can use Pi HATs on the Pico.

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.

YOU NEED

- **Pico or Pico W**
- **Pimoroni Drum HAT**
- **5x male-to-male jumper wires**
- **A large breadboard**
- **2x 10Kohm resistors)**
- **Code: https://github.com/lesp/LXF-304-PiTut1-Pico-DrumHAT-Shortcut-Keypad/archive/refs/heads/main.zip**

HATs only work on a Raspberry Pi, right? Generally yes, but sometimes they can work with the Pico. Pimoroni's Drum HAT is an older HAT with eight pads that can be used to perform drum solos and such. For this tutorial, we will repurpose a Drum HAT to be a USB human interface device (USB HID) via a Raspberry Pi Pico running CircuitPython.

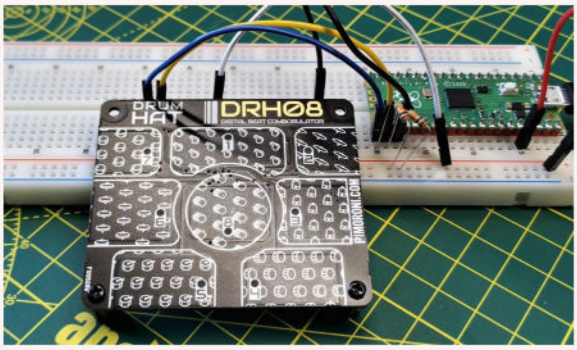
Drum HAT's CAP1188 capacitive touch sensor needs just four pin connections to our Pico. The problem is that the 20x2 (40-pin) configuration of the Raspberry Pi means we can't just drop it on to a breadboard. This is where some careful snipping of header pins is required. Using header pins, a long breadboard and some jumper wires, we can connect our two I2C pins GPIO16 to SDA, GPIO17 to SCL, 3V3 to 3V3 and GND to GND. The I2C pins each require a 10Kohm resistor from the pins to the 3V3 rail of the breadboard so the pins are pulled high. See the circuit diagram in the download for clear pin locations.

Prepare CircuitPython

Connect the Pico to your computer while holding the BootSel button. This forces the Pico into bootloader mode and a new drive, **RPI-RP2**, appears in the File Manager. Go to **https://circuitpython.org/board/raspberry_pi_pico/** for the Raspberry Pi Pico or **https://circuitpython.org/board/raspberry_pi_pico_w/** for the Pico W. Download the latest stable version of CircuitPython for your Pico. From the **Downloads** folder, copy the **CircuitPython** file (a UF2 firmware image) to the **RPI-RP2** drive. After a few moments, the drive disappears and is replaced with **CIRCUITPY**. This is where we will write the code for the project.

The next step is to copy a series of files into the **/lib** directory of **CIRCUITPY**. Go to **https://circuitpython.org/libraries** and download the library bundle for the version of CircuitPython you downloaded. We downloaded 8.05, so downloaded the bundle for CircuitPython 8. Extract the downloaded archive and navigate to that directory. From there we need to copy three directories – **adafruit_bus_device, adafruit_cap1188** and **adafruit_hid** – into the **lib** directory of **CIRCUITPY**. These libraries will be used in our code.

With the libraries copied, open your favourite text editor. We chose *Thonny* as it can work directly with CircuitPython devices. If you plan to use *Thonny,* check the boxout *(opposite)* for a quick setup guide.

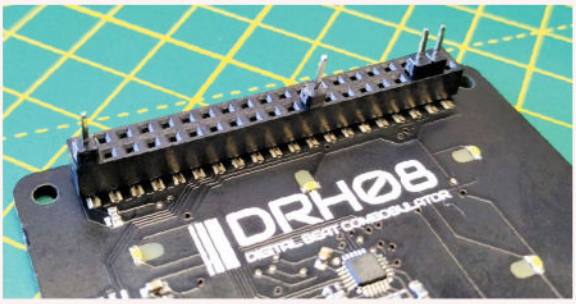


In not too many wires, and not too much time, you can reuse an old Raspberry Pi HAT with a Raspberry Pi Pico.

Create a new file and start the project code by importing a series of modules. The first is board, which enables our code to use the Pico's GPIO. Next, adafruit_cap1188.i2c is the module for the CAP1188 touch sensor. Hardware-accelerated bus access is provided via busio; time is used to control the speed of the code, and usb_hid emulates USB input devices.

```
import board
from adafruit_cap1188.i2c import CAP1188_I2C
import busio
import time
import usb_hid
```

Next is another series of imports, this time for setting up our Pico as a USB keyboard using a US layout. We also configure the device to use USB



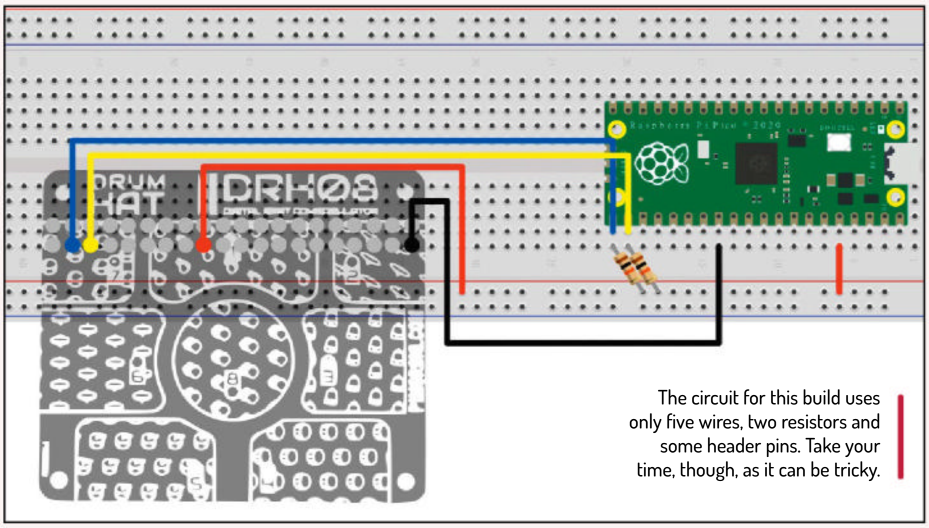
Make sure that the pins are spaced correctly and in the correct row. Yes, we got it wrong a few times.

keycodes that translate into what a keyboard would send to the computer. Finally we import consumer codes that are used by multimedia keyboards for extra functions (volume and so on).

```
from adafruit_hid.keyboard import Keyboard
from adafruit_hid.keyboard_layout_us import KeyboardLayoutUS
from adafruit_hid.keycode import Keycode
from adafruit_hid.consumer_control_code import ConsumerControlCode
from adafruit_hid.consumer_control import ConsumerControl
```

Create an object, **i2c**, to store the I2C pin setup. The Drum HAT is connected to GPIO16 (SDA) and GPIO17 (SCL).

```
i2c = busio.I2C(board.GP17, board.GP16)
```


The circuit for this build uses only five wires, two resistors and some header pins. Take your time, though, as it can be tricky.

Create an object, **cap**, and then call the CAP1188 module, passing the **i2c** object and the **i2c** address, **0x2c**. We found the address using this resource: **https://pinout.xyz/pinout/drum_hat**.

```
cap = CAP1188_I2C(i2c, 0x2c)
```

Next, create three objects. The first creates a USB keyboard that we can control. Then we set the keyboard layout to US. Finally, we enable an easier method to access ConsumerControl codes.

```
keyboard = Keyboard(usb_hid.devices)
layout = KeyboardLayoutUS(keyboard)
consumer = ConsumerControl(usb_hid.devices)
```

Create a **while True** loop to run the main code body.

```
while True:
```

Using a series of conditional tests, we will check for a button press. If pad 1 is pressed, the value will report **True** and trigger a sequence of code to run. First it will send the keycode for the B key, then print a message to the Python shell. All keypresses will be released and a short 0.3-second sleep will reduce the chance of multiple keypresses. Tweak this timing to suit.

```
    if cap[1].value:
        keyboard.send(Keycode.B)
        print("B pressed")
        keyboard.release_all()
        time.sleep(0.3)
```

We won't cover every pad on Drum HAT, as there are eight. We will pick out those that show different ways to send keyboard commands. That said, every Drum HAT pad is represented by **cap[x]**, where x is a number between 1 and 8, so it is easy to remember which pad does what. All the code is in the download.

For the second pad, we will send a keyboard command to move the cursor to the right using the right arrow key. As before, send the code, print a message, release the keys and then pause.

```
    if cap[2].value:
        keyboard.send(Keycode.RIGHT_ARROW)
        print("Right arrow pressed")
        keyboard.release_all()
        time.sleep(0.3)
```

The third pad will send a consumer code to mute the system audio. Consumer codes offer features that are not part of the standard keyboard specification; in this case, audio control. The process is the same. Send command, print a message, release keys, then pause.

```
    if cap[3].value:
        consumer.send(ConsumerControlCode.MUTE)
        print("MUTED")
        keyboard.release_all()
        time.sleep(0.3)
```

Skipping to the fifth pad, we send a raw keycode value that maps to a Super/Windows/Command key. Pressing this pad will trigger a Start menu to appear.

```
    if cap[5].value:
        keyboard.send(227)
        print("Super / Windows key pressed")
        keyboard.release_all()
        time.sleep(0.3)
```

Save code to **CIRCUITPY** as **code.py**. CircuitPython loads **code.py** each time it is power cycled. In *Thonny* click on Stop, then Run to restart the code. The Python shell prints each message as a key is pressed. The project can be used with Linux, Windows or Mac OS machines and appears as a basic USB keyboard.

» USING THONNY WITH CIRCUITPYTHON

Thonny is an excellent Python editor. It works with Python on our computer and it can work with MicroPython and CircuitPython on a microcontroller, such as the Raspberry Pi Pico. *Thonny* can be downloaded from **https://thonny.org** – once installed, go to Tools > Options, then click on the Interpreter tab and select CircuitPython (Generic). If we were using MicroPython on the Pico, we would select the Raspberry Pi Pico option; RP2040 is for other RP2040-based boards. Under Port, leave the option as Try To Detect Port Automatically. Click OK to save. Plug in your Raspberry Pi Pico running CircuitPython, and *Thonny* detects and connects to the device. If it fails, click Stop to retry. If it still fails, go back to the Interpreter menu and manually select the port. The Python shell (also called an REPL, Read, Evaluate, Print, Loop) starts and in there we can interactively run code on the Pico. The Python shell is also used to show the output of a project's code. We use it to print the current keypress as a debug method.

Inside BlendOS

Matt Holder discovers how to use the Arch-based BendOS to mash together multiple distributions. Does it give the best of them all?

What if you could mix the friendliness of Ubuntu, the cutting edge of Arch and the stability of Fedora? Would that be the ultimate distro? Welcome to BlendOS! This is an Arch-based distribution that contains many modern features and allows for applications to be run from three different distros (Ubuntu, Fedora and Arch). It sounded so interesting we decided to have an in-depth look, get you up and running and interview the project's founding developer.

Let's begin with a discussion about Arch. When we started using Linux, there was a mantra along the lines of: if you want to use a different operating system, use Linux, and if you want to learn about the depths of Linux, use Arch. Arch now provides a text-based installer tool that walks new users through the installation process. This is by no means as user-friendly as something like the Ubuntu installer, but it's a lot simpler than the traditional option, which is to carry out every stage manually using the relevant command-line tools. For example, to set the correct keyboard layout, you would need to run `ls /usr/share/kbd/keymaps/*/*.map.gz`, which lists the names of the keyboard mapping files. The next command to run sets the layout, which needs to be as follows `loadkeys de-latin1`. Some other steps that then need to be carried out involve configuring the Ethernet or Wi-Fi connection, partitioning the disk, copying files, using *chroot* to invoke commands on the new installation, setting the root password, adding a user and installing the bootloader. This can be incredibly daunting, but it's a very powerful way of learning what is involved and how the jigsaw pieces all fit together to end up with a working distribution.

Arch is renowned for having access to bleeding-edge software. The pace of software updates is so great that it isn't recommended to wait too long between upgrades, because if lots of packages change version, it can cause stability issues.

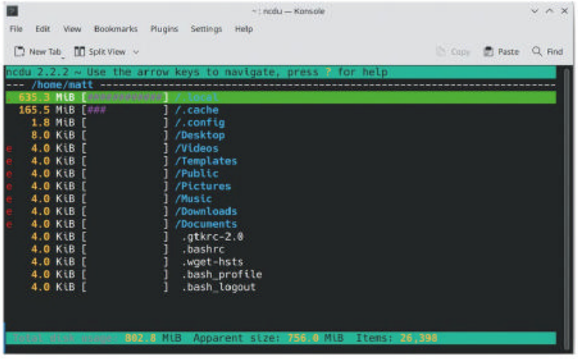
Around the blend

Now to BlendOS. It features a read-only or immutable root filesystem. This has been accomplished in an interesting way, with filesystem overlays available to install packages. These are mounted transparently on top of the root filesystem and the distribution sees the files as all being on the same filesystem. A subsequent boot allows for the overlay to be merged to the root filesystem, should this be required. As described when discussing other immutable operating systems, it is important to keep the root filesystem as standard as possible, so that extra security concerns are not added to the base operating system. Due to the sheer pace of change and reduced testing time, the base of Arch won't necessarily be as stable as something like Ubuntu or OpenSUSE's MicroOS.

A default BlendOS installation utilises the ext4 filesystem, which is very well tested, but doesn't have native support for filesystem snapshots. The author has provided a handy GUI that allows for the current state to be saved. It is also possible to roll back to the previous snapshot, just in case changes you make have broken the OS in some way. Note, this is not the same as something like a filesystem snapshot that can be carried out with ZFS or BTRFS. By default, BlendOS saves the state every 12 hours, but this can be disabled if required. During writing the GUI options for saving the system state were removed, hopefully these will be restored by the time you're reading this.

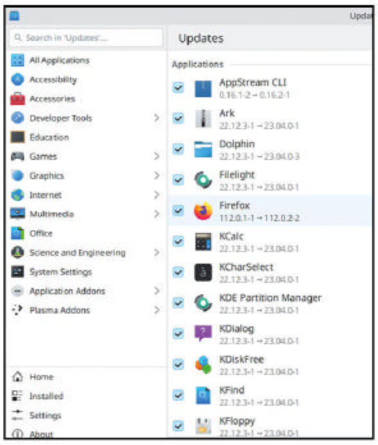
In the second major version of BlendOS, the author has integrated support for the *Waydroid* emulator. As you might guess, *Waydroid* requires a Wayland session to run and provides access to an emulated Android installation. The method used to achieve this is to run *Waydroid* within a container. Once *Waydroid* has been installed, it is possible to install an app store and any apps that are installed are displayed in the launcher.

Two versions of BlendOS are available, with the first using the Gnome desktop environment and the second running KDE. Gnome favours a more modern paradigm, with multiple workspaces and a single window per workspace. It is argued that this allows for better focus on a particular task. We find it difficult to use when trying to compare two documents side by side. KDE has a more traditional paradigm and looks a lot like using a Windows device. KDE, however, is infinitely customisable and has a huge number of extensions and plugins that enable you to turn KDE into anything you want it to be. The screenshots here are of the KDE version of blendOS, but installing the Gnome edition should still be relatively simple.



```
ncdu 2.2.2 ~ Use the arrow keys to navigate, press ? for help
--- /home/matt -----------------------------------------------------------------
. 635.3 MiB [############] /.local
  165.5 MiB [###         ] /.cache
    1.8 MiB [            ] /.config
    8.0 KiB [            ] /Desktop
e   4.0 KiB [            ] /Videos
e   4.0 KiB [            ] /Templates
e   4.0 KiB [            ] /Public
e   4.0 KiB [            ] /Pictures
e   4.0 KiB [            ] /Music
e   4.0 KiB [            ] /Downloads
e   4.0 KiB [            ] /Documents
    4.0 KiB [            ]  .gtkrc-2.0
    4.0 KiB [            ]  .bashrc
    4.0 KiB [            ]  .wget-hsts
    4.0 KiB [            ]  .bash_profile
    4.0 KiB [            ]  .bash_logout


 Total disk usage: 802.8 MiB  Apparent size: 756.0 MiB  Items: 26,398
```

The NCurses Disk Utility (NCDU) has been installed directly on the host operating system, using an overlay.



Updates

Applications

☑ AppStream CLI 0.16.1-2 → 0.16.2-1
☑ Ark 22.12.3-1 → 23.04.0-1
☑ Dolphin 22.12.3-1 → 23.04.0-3
☑ Filelight 22.12.3-1 → 23.04.0-1
☑ Firefox 112.0.1-1 → 112.0.2-2
☑ KCalc 22.12.3-1 → 23.04.0-1
☑ KCharSelect 22.12.3-1 → 23.04.0-1
☑ KDE Partition Manager 22.12.3-1 → 23.04.0-1
☑ KDialog 22.12.3-1 → 23.04.0-1
☑ KDiskFree 22.12.3-1 → 23.04.0-1
☑ KFind 22.12.3-1 → 23.04.0-1
☑ KFloppy 22.12.3-1 → 23.04.0-1

Operating system updates are installed from the KDE Discover app.

As described above, it is possible to install to the base OS. Packages are installed as an overlay, which is merged into the root filesystem at the next boot. Unlike with MicroOS, where a specific command is needed to install to the base OS, the standard Arch commands can be used. For example, to install the *ncurses* disk usage tool, run the following command: `sudo pacman -S ncdu`. Having said that, this should be used as sparingly as possible. A better way to install apps is to use Flatpaks from the *Discover* software tool. See the walkthrough *(page 53)* to learn how to install an app. Using Flatpaks means that apps are sandboxed from the base OS and potentially from each other.

Keep it contained

An integral part of BlendOS is the ability to install containers and apps within them. This allows for a best-of-all worlds approach, where apps can be

QUICK TIP

BlendOS can be downloaded from https://github.com/blend-os/blendOS/releases

» INTERVIEW WITH BRENDON CHUNG

Q What caused you to become interested in Linux and what was the first distro you used?

A I started using Linux at the age of seven. I switched to it as my old laptop was slowing down with a third-gen i5. Also, since I used to develop Android apps using Java and XML *(Android Studio)* back then, Android app builds were really slow on Windows. That's when I decided to try Linux on my laptop. Arch was the first distro I used, followed by Ubuntu.

Q What is the biggest change that you've seen in Linux since starting working with it? We can recall editing XF86 config files and dreading damaging the monitor!

A I have been using Linux since around 2017, when I was seven. In those six to seven years, I've seen Linux evolving and becoming more popular and user-friendly. Back in 2017, Ubuntu was really popular, with most other distributions that are now considered user-friendly being branded as niche. I've witnessed the rise of immutable and atomic distributions, as well as the huge strides in game support on Linux distributions. Containerised packages have become the norm for Linux distributions (apart from Arch, of course).

Q What was the first public project that you were involved with?

A My first popular public project was Ubuntu Unity (**https://twitter.com/ubuntu_unity**), which I released in May 2020 and is now an official Ubuntu flavour, although I had made some distros and projects before that as well.

Q How do you differentiate BlendOS from something such as Vanilla OS?

A BlendOS is aimed at a general computer user who wants to be able to run their regular desktop apps (irrespective of Linux distribution, thus blurring the boundaries), apps they use on their Android phones, as well as web apps from a curated list. Vanilla OS is a great distro, but our goals are completely different. As per the Vanilla OS website, it aims to deliver a stable, Debian-based immutable distribution for Linux users. Besides, BlendOS offers support for multiple DEs and is going to provide the ability to create remixes. We're also working on server and gaming editions for the next major release of BlendOS.

BlendOS uses Arch Linux as its base and allows you to install packages using the system package manager while retaining immutability and atomicity (using overlays), without having to reboot, unlike other immutable operating systems. Also, unlike Vanilla OS, which only exports apps that have been installed using *Apx*, which is its package manager (and its implementation of binary availability doesn't work for some apps, especially if the

»

» ARCH USER REPOSITORY

One of the many things that Arch is praised for is the Arch User Repository, commonly known as the AUR. This is a repository of software that is maintained by a large number of community members and has support for some of the most bleeding-edge software available. The AUR would be an excellent thing to enable within an Arch container, so that the newest software can be experienced without it being installed directly on the host OS.

A number of steps are required to install the AUR, which can be accomplished by opening the BlendOS *Settings* app, clicking on the play icon on an Arch container, and running the following commands:

```
$ sudo pacman -S --needed base-devel git
$ git clone https://aur.archlinux.org/yay.git
$ cd yay
$ makepkg -si
```

Applications can be searched for using the following command – in this example, *Google Chrome* is searched for and installed:

```
$ yay -Ss google-chrome
```

Once the name of the package has been determined from the search, it can be installed as follows:

```
$ yay -S google-chrome
```

Packages can be uninstalled using `yay -Rns google-chrome`.

installed from some of the most popular distros – Ubuntu and Fedora. This has been discussed in previous articles covering Fedora Silverblue, OpenSUSE's MicroOS and Vanilla OS. When apps are installed within a container, they are tightly integrated with the host OS. In the case of BlendOS, this means the app can be opened from the launcher and the command line. If *Audacity* is installed in a container, it can be run by entering `audacity` at the host OS's command line. During the writing of this article, the container prioritisation feature was changed. Originally containers could be assigned priorities, so that if the same app is installed in multiple containers, the one with highest priority is invoked by default. After updates, this prioritisation feature was changed to allow for apps to be mapped to a particular container. For example, *Audacity* could be assigned to the Ubuntu container and *Fritzing* could be assigned to the Fedora container. When a containerised app has been opened, it automatically has access to the host's filesystem, so it can be used as though it were installed directly.

QUICK TIP
More information can be found at https://blendos.co/blend-os-v2/

See the walkthrough *(page 53)* to discover how to install containers. Apps can be installed to them by navigating to the BlendOS *Settings* tool, selecting the Containers tab and pressing the play button next to the container name. This opens a terminal on the selected distro. The *Apt* tool can be used in an Ubuntu container with `apt search NAME` to search for packages and `apt install NAME` to install them. Within Fedora, instead of `apt` use `dnf`. Finally, within an Arch container, `pacman` can be used. See the boxout *(left)* to learn how to use the Arch User Repository (AUR).

At the time of writing, installing in a dual or multi-boot environment was possible, but not easy. When selecting the partitions to create, format and write to, the data was copied over successfully, but the *GRUB* bootloader was not installed. Performing these missing steps is outside of the scope of this tutorial. The steps section suggests using the entire hard drive and installing on a spare device or a virtual machine. *Waydroid* support is only possible on a physical device.

BlendOS is a relatively new endeavour, started by a teenager, Brendon Chung. Given the early state of the project, the tools and level of integration between containers, filesystem layers and all of the other items discussed is incredibly impressive. Chung also maintains Ubuntu Unity Remix, which re-integrates the Unity 7 codebase, and is heavily involved in Unity 7 development and Unity on Arch. He works on several other projects, too.

We hope you enjoy using blendOS and learning more about all the technologies involved. LXF

» INTERVIEW WITH BRENDON CHUNG

package name is different from the binary or app name), any apps or binaries installed in a container instantly appear on the host in BlendOS. This makes it a true blend of distributions. Again, BlendOS gives you the flexibility to use any apps or binaries from any container, in spite of a clash.

Q You're involved in Ubuntu Unity. What made you decide to create a distribution using Unity as the desktop environment?

A I had used Ubuntu 17.04 back in 2017 and really loved Unity 7. So, when Unity 7 was discontinued, I wanted to bring it back. I created Ubuntu Unity to give Unity 7 a new life.

Q Can you share a few words about some of your other projects?

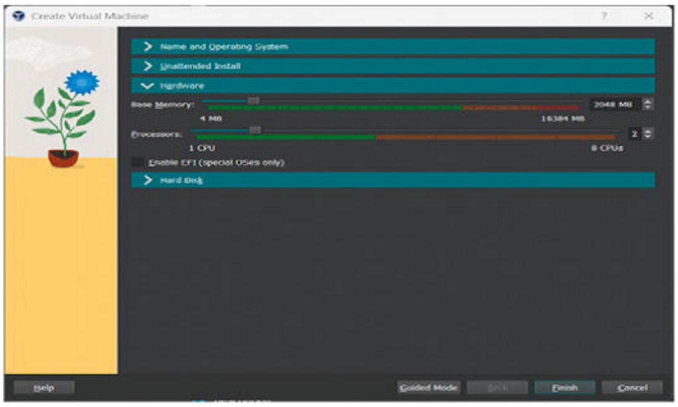
A Sure. You already know about Ubuntu Unity, which is an official Ubuntu flavour. That was one of my first popular projects that I created about three years ago. We had a new release recently (23.04) and next up is the Lomiri edition of Ubuntu Unity. I released Unity 7.7 and Unity X at the beginning of this year. I had also ported Unity 7.6 to Arch late last year and will be porting Unity 7.7 to Arch, too. I had released the Manjaro edition of Unity a few months ago, and it's time to update that to use Unity 7.7 as well.

Some other projects I've created and maintain include BlendOS, Ubuntu Web (a Chrome OS alternative, which will be a variant of BlendOS), *Gamebuntu* (a tool to simplify gaming on Ubuntu), *una* (similar to *yay* or *paru;* a client for the MPR, equivalent to the AUR but for Debian-based distros), lol (an alternative for the Ubuntu Snap Store), Modren (a store that supports Snaps, Flatpaks, *Apt* packages and the direct installation of DEBs), Skull (an old project of mine that planned to support Linux phones with a UEFI, without the need for an Android base or Halium), and *Ubuntu Remixes,* which is a tool to create Ubuntu-based distros.

Q Which project do you enjoy being involved with the most and why?

A I really enjoy working on Unity 7 and BlendOS. They have really active communities and are expected to grow greatly.

INSTALL AND CONFIGURE BLENDOS



Create Virtual Machine

Name and Operating System

Unattended Install

Hardware

Base Memory: 2048 MB (4 MB – 16384 MB)

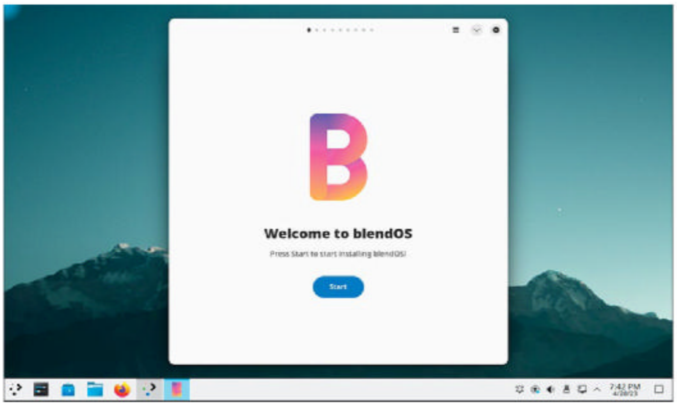
Processors: 2 (1 CPU – 8 CPUs)

☐ Enable EFI (special OSes only)

Hard Disk

1 Create a virtual machine

Navigate to the BlendOS website and download the ISO file. Open your virtualisation system of choice and create a new virtual machine. For testing, 2GB RAM, two vCPUs and 25GB of disk space will suffice. Attach the ISO file and switch on the virtual machine that you have created.

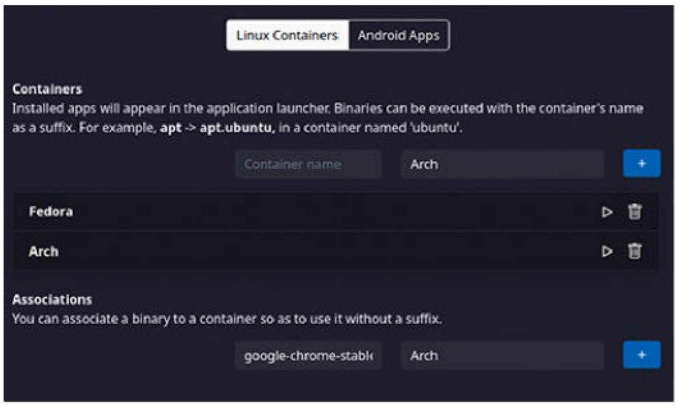


Welcome to blendOS

Press Start to start installing blendOS!

2 Set keyboard and language

Once the VM has loaded, follow through the steps on the installer to select the relevant keyboard layout, language, define a new user and select the drive to install to. Make sure you use the entire disk. To select the language, search for it, find the relevant one and select Main, then select the language itself.



Containers

Installed apps will appear in the application launcher. Binaries can be executed with the container's name as a suffix. For example, **apt** -> **apt.ubuntu**, in a container named 'ubuntu'.

Container name: Arch

Fedora

Arch

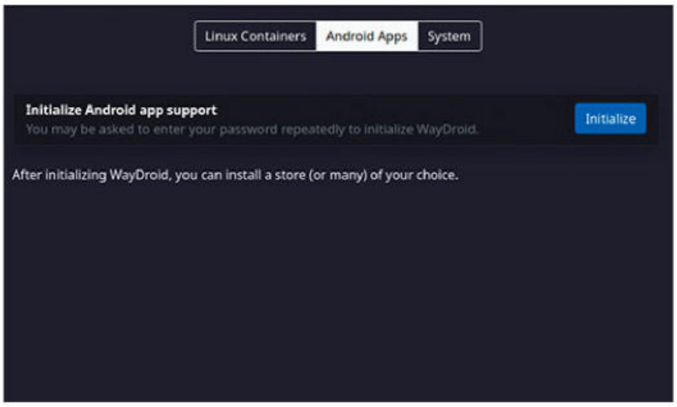
Associations

You can associate a binary to a container so as to use it without a suffix.

google-chrome-stable: Arch

3 Create a container

Open the BlendOS *Settings* app, enter a name, select the container type and click Create Container. Associations can be made between a binary and a container. This means that it is easy to run a command in a certain container, even if the same application is installed in multiple ones.



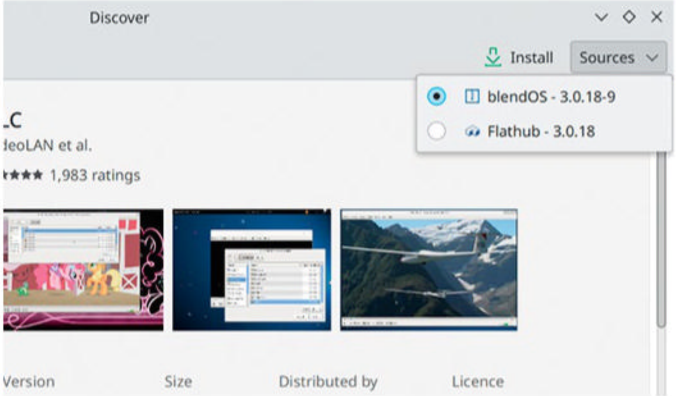
Initialize Android app support

You may be asked to enter your password repeatedly to initialize WayDroid.

After initializing WayDroid, you can install a store (or many) of your choice.

4 Install app stores

From the BlendOS *Settings* app, select the Android Apps tab and click Initialize. Once installed, install either or both app stores, which can then be opened to install apps. Once apps have been installed, they can be opened from the launcher. Apps open as their own window, alongside native apps.

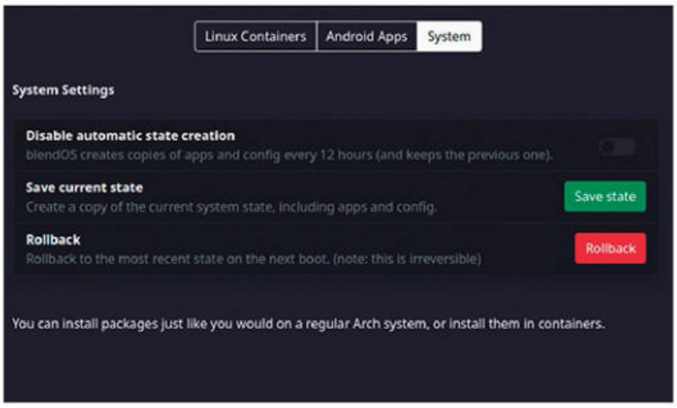


● blendOS - 3.0.18-9

○ Flathub - 3.0.18

5 Add some apps

Discover is KDE's software store. Open this from the launcher and search for the application you wish to install. Notice in the top-right corner that some apps have two sources. Flatpaks are sandboxed and BlendOS installs to an overlay, which is merged to the immutable root at next boot.



System Settings

Disable automatic state creation

blendOS creates copies of apps and config every 12 hours (and keeps the previous one).

Save current state

Create a copy of the current system state, including apps and config.

Rollback

Rollback to the most recent state on the next boot. (note: this is irreversible)

You can install packages just like you would on a regular Arch system, or install them in containers.

6 Save state and rollback

The System tab in the BlendOS *Settings* application shows controls for saving the state of the system, which can be carried out after any operation and this process occurs every 12 hours automatically. The automated operation can be switched off and rollback to the previous state can be invoked.

TUIFEED

Credit: https://github.com/veeso/tuifeed

Minimalist RSS reader

As far as **Shashank Sharma** is concerned, it's not social media but ads that are the bane of modern internet usage, and he has a clever workaround.

OUR EXPERT

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

For voracious consumers of web content, the greatest obstacle in the quest for information is the need to switch from one website to another. Wouldn't it be great if you could browse through hundreds of your favourite internet resources with a single app or utility? That's the gift of RSS feeds.

With the minimalist feed reader *Tuifeed,* you can quickly scroll through the content of all your favourite sites without once having to reach for the browser.

Unfortunately, *Tuifeed* and most of its ilk, such as *Nom (see box below),* are limited to serving text-based content from websites. Some graphical alternatives, however, can go a step further and also fetch podcasts.

Written in Rust and released under the MIT licence, the project isn't available in the repositories of desktop distros, but installation is still quite straightforward.

If you don't already have Rust and *Cargo* installed, you can use your distro's package manager to install them. The `sudo dnf install rust cargo` command installs them on RPM-based distributions, such as Fedora. You can similarly run `sudo apt install cargo`, which also installs the **Rustc** package, if you're running Ubuntu or Debian, or one of their derivatives.

You can now use *Cargo,* the package manager for Rust packages, to install *Tuifeed* with the `cargo install tuifeed` command. You can alternatively install the latest version using the official shell script:

```
curl --proto '=https' --tlsv1.2 -sSLf "https://git.io/
J1OOZ" | sh
```

Working with feeds

Tuifeed claims to support both RSS and Atom feeds. You must add the feeds you're interested in to the **~/.config/tuifeed/config.toml** file. Open the file in your favourite text editor or run the `tuifeed -c` command.

All your feeds are added under the [sources] section of the **~/.config/tuifeed/config.toml** file with the `"Website Name" = "URL"` format:

```
[sources]
"Comicosity" = "https://www.comicosity.com/feed/"
"TechRadar Software" = "https://www.techradar.com/
rss/news/software"
"Techradar" ="https://www.techradar.com/rss"
"Ars Technica Tech Policy" = "https://feeds.
arstechnica.com/arstechnica/tech-policy"
"Ars Technica" = "https://feeds.arstechnica.com/
arstechnica/index"
"IPLeaders Blog" = "https://blog.ipleaders.in/feed"
"SpicyIP" = "https://spicyip.com/feed"
"Best Thrillers" = "https://bestthrillers.com/feed"
```

After you have added the feeds for all of your favourite internet resources into the **~/.config/tuifeed/**

» WORKING WITH NOM

Nom (**https://github.com/guyfedwards/nom**) is every bit as minimalist as *Tuifeed,* if not more so. It doesn't have a paned interface, or any colours for that matter. Unlike *Tuifeed,* however, *Nom* never messes up the content of an article and respects paragraphs.

Written in Go, you can install *Nom* with the `go install github.com/guyfedwards/nom/cmd/nom@latest` command. You can send feeds to the **~/.config/nom/config.yml** file:

```
feeds:
url: https://feeds.arstechnica.com/arstechnica/tech-policy
name: ArsTechnica Tech Policy
- url: https://feeds.arstechnica.com/arstechnica/index
name: ArsTechnica
- url: https://bestthrillers.com/feed
name: Best Thrillers
```

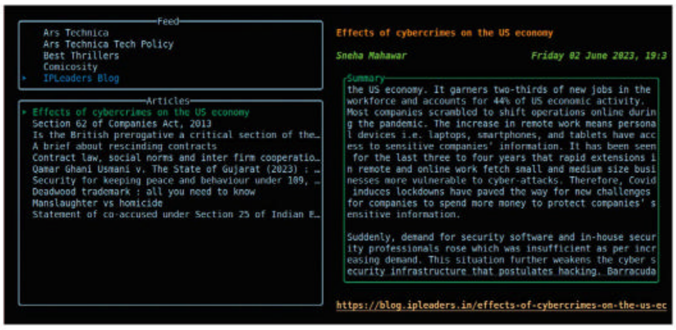
You can run `nom --feed URL` to test how it displays the specified feed without adding it.

When you run *Nom* after adding all the feeds to the config file, the project displays a numbered list of all the fetched articles, along with the feed's name:

```
11. TechRadar Software: Nvidia's latest AI model
12. TechRadar Software: Microsoft gives one of
14. TechRadar Software: Microsoft is finally
74. ArsTechnica: CEO: Raspberry Pi stock to
```

We've only featured a truncated list to showcase how *Nom* displays the content from the different feeds. Use the arrow keys or PgUp/PgDn to move through the list.

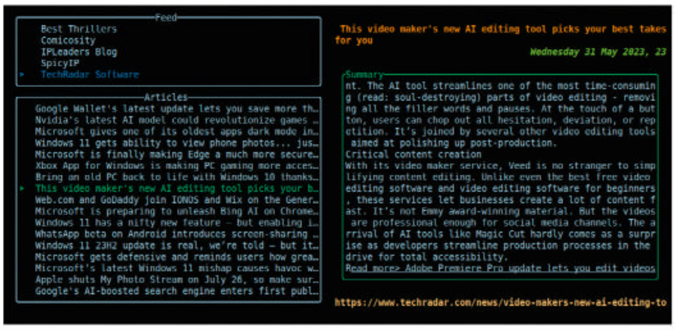
To read an article, select it from the list and press Enter. Use the up/down keys to read through the article and press Esc to move back to the article list.



```
Feed
    Ars Technica
    Ars Technica Tech Policy
    Best Thrillers
    Comicosity
 ▸  IPLeaders Blog

Articles
▸ Effects of cybercrimes on the US economy
  Section 62 of Companies Act, 2013
  Is the British prerogative a critical section of the…
  A brief about rescinding contracts
  Contract law, social norms and inter firm cooperatio…
  Qamar Ghani Usmani v. The State of Gujarat (2023) : …
  Security for keeping peace and behaviour under 109, …
  Deadwood trademark : all you need to know
  Manslaughter vs homicide
  Statement of co-accused under Section 25 of Indian E…
```

```
Effects of cybercrimes on the US economy

Sneha Mahawar                        Friday 02 June 2023, 19:3

Summary
the US economy. It garners two-thirds of new jobs in the
workforce and accounts for 44% of US economic activity.
Most companies scrambled to shift operations online durin
g the pandemic. The increase in remote work means persona
l devices i.e. laptops, smartphones, and tablets have acc
ess to sensitive companies' information. It has been seen
 for the last three to four years that rapid extensions i
n remote and online work fetch small and medium size busi
nesses more vulnerable to cyber-attacks. Therefore, Covid
 induces lockdowns have paved the way for new challenges
for companies to spend more money to protect companies' s
ensitive information.

Suddenly, demand for security software and in-house secur
ity professionals rose which was insufficient as per incr
easing demand. This situation further weakens the cyber s
ecurity infrastructure that postulates hacking. Barracuda

https://blog.ipleaders.in/effects-of-cybercrimes-on-the-us-ec
```

You'll have to play around with your terminal emulator's colour scheme to tweak Tuifeed's appearance because it doesn't support themes.



```
Feed
    Best Thrillers
    Comicosity
    IPLeaders Blog
    SpicyIP
 ▸  TechRadar Software

Articles
  Google Wallet's latest update lets you save more th…
  Nvidia's latest AI model could revolutionize games …
  Microsoft gives one of its oldest apps dark mode in…
  Windows 11 gets ability to view phone photos... jus…
  Microsoft is finally making Edge a much more secure…
  Xbox App for Windows is making PC gaming more acces…
  Bring an old PC back to life with Windows 10 thanks…
▸ This video maker's new AI editing tool picks your b…
  Web.com and GoDaddy join IONOS and Wix on the Gener…
  Microsoft is preparing to unleash Bing AI on Chrome…
  Windows 11 has a nifty new feature – but enabling i…
  WhatsApp beta on Android introduces screen-sharing …
  Windows 11 23H2 update is real, we're told – but it…
  Microsoft gets defensive and reminds users how grea…
  Microsoft's latest Windows 11 mishap causes havoc w…
  Apple shuts My Photo Stream on July 26, so make sur…
  Google's AI-boosted search engine enters first publ…
```

```
 This video maker's new AI editing tool picks your best takes
for you
                                 Wednesday 31 May 2023, 23

Summary
nt. The AI tool streamlines one of the most time-consumin
g (read: soul-destroying) parts of video editing - removi
ng all the filler words and pauses. At the touch of a but
ton, users can chop out all hesitation, deviation, or rep
etition. It's joined by several other video editing tools
 aimed at polishing up post-production.
Critical content creation
With its video maker service, Veed is no stranger to simp
lifying content editing. Unlike even the best free video
editing software and video editing software for beginners
, these services let businesses create a lot of content f
ast. It's not Emmy award-winning material. But the videos
 are professional enough for social media channels. The a
rrival of AI tools like Magic Cut hardly comes as a surpr
ise as developers streamline production processes in the
drive for total accessibility.
Read more> Adobe Premiere Pro update lets you edit videos

https://www.techradar.com/news/video-makers-new-ai-editing-to
```

With the summary pane selected, you can also hit Enter to open the selected article in the default web browser on your desktop.

config.toml file, you can launch *Tuifeed* with the `tuifeed` command.

The interface is split into three panes. All the feeds you've added to the **config.toml** file are listed in the top-left under the Feed pane. Below this is the Articles pane, which features the latest stories from the selected feed.

The right-hand side of the interface is reserved for the selected article. You'll find the title, author name and publication date at the top, and the actual content of the article under the Summary pane.

When you first launch *Tuifeed*, the Feed pane is selected. You can use the up and down arrow keys to navigate the Feed pane and select the feed you're interested in. The selected feed is highlighted and can also be identified with the `>` symbol on the extreme left. To reload a feed, select it from the list and press `r`. You can alternatively press Ctrl+r to reload all the added feeds.

You can press the right arrow key or Tab to move to the Articles pane. Once again, the up/down arrow keys can be used to scroll through the list of articles within the selected feed. The content of each article is displayed in the right half of *Tuifeed's* interface.

To read an article, select it from the Articles pane, and then press the right arrow key to move to the Summary pane. You can now scroll through the article using the up/down arrow keys.

From the Summary pane, you can use the left arrow key to move to the Articles pane. Just press the left arrow key again to move up to the Feed pane.

Possibly the greatest advantage of reading content using feed readers such as *Tuifeed* is that they automatically filter the ads out, providing for a smooth reading experience.

Tuifeed doesn't offer too many configurable parameters. It has no themes, fonts, colours or other tweaks to improve your reading experience. You can, however, choose to display the author name and the timestamp for the published articles. To do so, add the following section to the **~/.config/tuifeed/config.toml** file below the [sources] section:

```
[article-title]
show-author = true
show-timestamp = true
```

This adds the name of the author and the timestamp for each article ahead of the title under the Articles pane. We suggest that you set these to false. These details can be seen on the right side of the *Tuifeed* interface above the Summary pane when you select an article from the Articles pane.

To quit *Tuifeed*, press Esc and then press Enter at the `'Are you sure you wish to quit?'` prompt. When you relaunch *Tuifeed*, it automatically fetches the latest stories and articles from the defined feeds.

Caveat emptor

Most feed readers boast a series of useful features such as the ability to bookmark favourite feeds, mark feeds or articles as read, open the article or a link within in a web browser, and so on.

You won't find any of these features in the minimalist *Tuifeed*. Adding a series of RSS feeds from different websites and displaying the latest stories from each is the extent of the project's capabilities.

But that's not all. As wonderful as *Tuifeed's* three-pane interface is, it isn't without problems. For one, the project struggles with displaying RSS 1.0 and Atom feeds. While it won't complain if you add such feeds to be fetched, the project can only download the article's title from such feeds, and not the actual content.

Also, because *Tuifeed* doesn't let you resize the individual panes, you have to make do with the limited reading area to scroll through the entire article. Depending on the formatting on the source website, *Tuifeed* either neatly breaks the content into individual paragraphs, or displays the whole article text as a single paragraph. There's no discernible pattern to *Tuifeed* rendering content in such a fashion.

Although RSS never achieved the level of adoption many thought it would, there are plenty of feed readers on offer. Your web browser, and maybe even preferred email client tool, probably has an RSS reader built in or available through extensions.

But if it's CLI tools you're interested in, refer to the box for a discussion of *Nom*, an incredibly robust CLI alternative that's just as barebones as *Tuifeed* but more adapt at displaying content. Or try your hands at *Newsboat*, a far more feature-rich alternative that we previously covered in **LXF255**.

If you're unsure what sites to add to your reading list, or are looking to move beyond your favourites, head to **https://blog.feedspot.com/rss_directory/** for a sizeable collection of feeds from hundreds of websites sorted into categories such as health, travel, fitness, law, fiction and so on for your reading pleasure. LXF

QUICK TIP

Because Tuifeed doesn't support resizing the individual panes, we suggest you maximise the terminal emulator window to full-screen to get the most out of Tuifeed's interface.

TIMESHIFT

Credit: https://github.com/teejee2008/timeshift

Keep your Linux install backed up

Nick Peers reveals the tools, techniques and tips you need to protect everything from personal files to your entire Linux installation.

OUR EXPERT

Nick Peers has been an ardent backup fanatic since spending one Christmas holiday manually recovering hundreds of text files one at a time from the command line back in the late '90s.

Do you need reminding of the importance of backups? From protecting your irreplaceable data to ensuring system glitches are mere annoyances rather than catastrophes, everyone needs a good backup regimen. In this tutorial we'll explore the best tools on offer from Ubuntu and Mint to provide everything you need, and we'll cover both backup and restore scenarios, whether reverting individual files to earlier versions or undoing a recent botched update or unwanted app installation. You'll never endure the stomach-churning pain of data loss again.

Home folder

The most important part of any backup regimen is to protect your personal files, from documents to photos and videos. Unlike your OS or apps, which can be reinstalled, your personal files can't easily be recreated – if at all. Both Ubuntu and Mint ship with tools to provide the means to automatically back up these files.

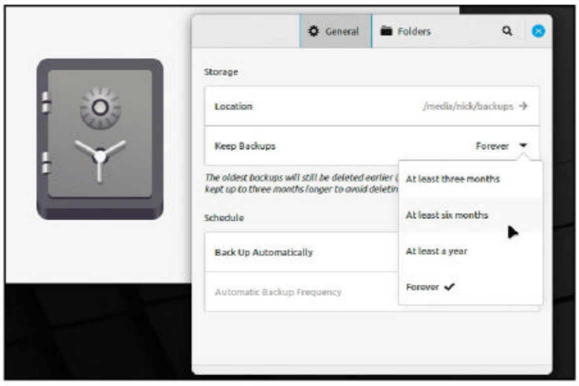
Mint's file backup tool is covered in the first box *(opposite)*, but Ubuntu users are better served by their file backup tool: *Backups*. Access it by typing `backups` into the launcher's search tool. It's actually a rebadged and integrated version of *Déjà Dup Backups*, and Mint users can add this via *Software Manager* (search for 'deja' and choose the System Package version). After installation, you'll find it under the label Backups.

Once launched, click Create My First Backup. Your **Home** folder is selected by default under Folders To Back Up, while **Downloads** and **Rubbish Bin** are excluded under Folders To Ignore. Use the + buttons to add more to either list before clicking Forward.

Next, you're prompted to select a location for your backup. In addition to any locally attached storage, there are options to back up directly to Google Drive (just enter your credentials when prompted), or you can back up to any network drive (click the ? button for the syntax to use, such as `smb://computer-name/folder`. Once selected, click Forward again.

Next, you're given the option of encrypting your backup (which is compressed into a tar.gz file) using a password before being returned to the main screen. From here, click Back Up Now to take your first backup.

You'll also see a Back Up Automatically switch – you can find the same switch under Preferences > General, which is accessed via the menu button. Beneath this is an Automatic Backup Frequency option, which can be toggled between Daily and Weekly as required.



Storage

Location /media/nick/backups

Keep Backups Forever

The oldest backups will still be deleted earlier i
kept up to three months longer to avoid deletin

At least three months
At least six months
At least a year
Forever ✓

Schedule

Back Up Automatically

Automatic Backup Frequency

Once you've configured your backup using the Backups tool, changes can be made by clicking the menu button followed by Preferences.

The General tab is also where you can change the destination of your backups, plus set a backup retention scheme, with backups kept for ever by default (basically until you run low on disk space), or kept for a minimum of three, six or 12 months. Finally, to change what's backed up, switch to the Folders tab.

Restoring backups

When your backup runs, any new files are added and changed files are recorded, too. They don't overwrite the original backed-up files, which means that not only can you restore the latest version of any backed up file, but you can roll back changes on individual files, too.

Should you need to restore any files from your backup, open *Backups* again, but this time switch to the Restore tab and wait while your backups are scanned. Once the initial scan is complete, you'll see a Files-like view of all the folders stored in the backup.

In the bottom-right is a Date field that displays the date (and time if more than one backup exists on a specific date) of the selected backup. If you're looking to restore an earlier version of a file, use this drop-down to select the date/time of the backup you want.

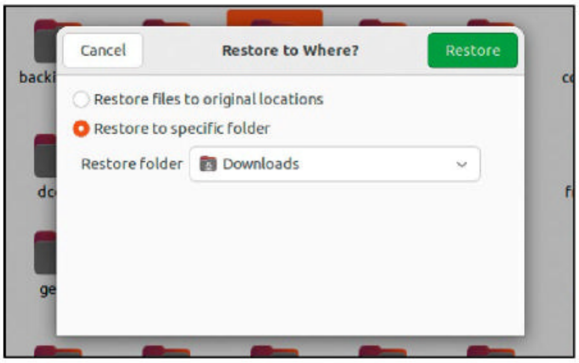
Backups requires you to manually select the folders or files to restore. To restore everything, press Ctrl+A followed by Restore; otherwise browse through the

QUICK TIP

Mint users will notice the version of Déjà Dup Backups offered in the main repositories isn't the latest – that said, there's little difference between it and the rather bloated 900MB-plus Flatpak version, which is why we recommend you install the older System Package version instead.

folders until you find what you need. Select multiple files and folders using Ctrl-click if necessary.

You're given the option of restoring your selection to their original locations, overwriting anything that's already present, or you can opt to restore to a different folder, which creates a separate copy of your restored files in a location of your choosing. Whatever you choose, click Restore, enter the encryption password if necessary, and finally click Forward to bring them back.



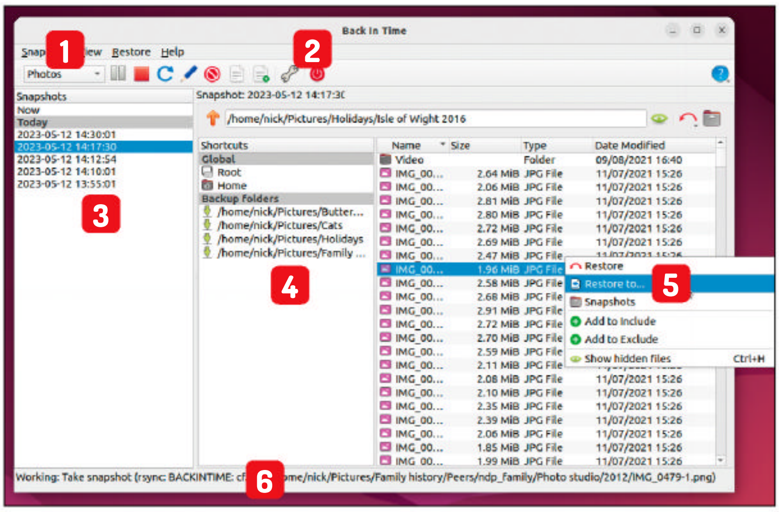
○ Restore files to original locations
● Restore to specific folder
Restore folder: Downloads

If you plan to restore an earlier version of a file, hedge your bets by copying it to a separate location for a thorough comparison.

Different approach

Backups is fine for basic backup tasks, but it has two weaknesses: you can only schedule daily or weekly backups, and you can only have a single backup job. Both shortcomings can be remedied with a tool called *Back In Time,* which enables you to update backups as often as every five minutes, plus lets you set up multiple profiles, each one enabling you to back up different folders with their own schedule, settings and destination. Visit **https://github.com/bit-team/backintime** to install the latest version – it's relatively simple to use, as the annotation *(above-right)* reveals.

NAVIGATE BACK IN TIME



Snapshot: 2023-05-12 14:17:3(

/home/nick/Pictures/Holidays/Isle of Wight 2016

Name	Size	Type	Date Modified
Video		Folder	09/08/2021 16:40
IMG_00...	2.64 MiB	JPG File	11/07/2021 15:26
IMG_00...	2.06 MiB	JPG File	11/07/2021 15:26
IMG_00...	2.81 MiB	JPG File	11/07/2021 15:26
IMG_00...	2.80 MiB	JPG File	11/07/2021 15:26
IMG_00...	2.72 MiB	JPG File	11/07/2021 15:26
IMG_00...	2.69 MiB	JPG File	11/07/2021 15:26
IMG_00...	2.47 MiB	JPG File	11/07/2021 15:26
IMG_00...	1.96 MiB	JPG File	[illegible]
IMG_00...	2.58 MiB	JPG File	[illegible]
IMG_00...	2.68 MiB	JPG File	[illegible]
IMG_00...	2.91 MiB	JPG File	[illegible]
IMG_00...	2.72 MiB	JPG File	[illegible]
IMG_00...	2.70 MiB	JPG File	[illegible]
IMG_00...	2.59 MiB	JPG File	[illegible]
IMG_00...	2.11 MiB	JPG File	[illegible]
IMG_00...	2.08 MiB	JPG File	11/07/2021 15:26
IMG_00...	2.10 MiB	JPG File	11/07/2021 15:26
IMG_00...	2.35 MiB	JPG File	11/07/2021 15:26
IMG_00...	2.39 MiB	JPG File	11/07/2021 15:26
IMG_00...	2.06 MiB	JPG File	11/07/2021 15:26
IMG_00...	1.85 MiB	JPG File	11/07/2021 15:26
IMG_00...	1.99 MiB	JPG File	11/07/2021 15:26

Working: Take snapshot (rsync: BACKINTIME: c[illegible]me/nick/Pictures/Family history/Peers/ndp_Family/Photo studio/2012/IMG_0479-1.png)

1 Profile Back In Time splits different jobs into profiles, each with its own backup source(s), destination, schedule and other settings.

2 Settings Click this button to set up new profiles and reconfigure existing ones – say by adding more folders or changing the backup schedule.

3 Timeline Use the Timeline pane to view and browse snapshots by date and time. Snapshots can be renamed using the pencil icon.

4 Backup folders All selected backup folders are listed here to aid navigation. You can also manually navigate using the Address bar above.

5 Browse and restore Right-click a folder or file in the Files pane to restore it to its original location or make a fresh copy in a new folder.

6 View progress In addition to this progress bar, you can monitor a backup's progress via the Back In Time menu bar icon.

System approach

System snapshots are designed to back up system files, enabling you to roll your system back to a specific point in time for whatever reason, but typically because you've run into a problem after installing a new app or update. By far the best tool for the job is *Timeshift,* which comes bundled by default in Linux Mint, and can be installed separately in Ubuntu.

Although the default Ubuntu repositories include *Timeshift,* the version offered is frozen at the point of your Ubuntu release – in the case of Ubuntu 22.04, that's version 21.09.1. If you'd prefer to keep *Timeshift* up to date in Ubuntu (the current version at time of writing is 22.11.2), install it through its own repository:

```
$ sudo add-apt-repository -y ppa:teejee20to08/timeshift
$ sudo apt-get update
$ sudo apt-get install timeshift
```

Timeshift can be launched through the Mint or Ubuntu launcher – indeed, you may have already set up *Timeshift* when you first installed Mint, as System Snapshots is offered during the initial setup process.

Configuring Timeshift

The first time you open *Timeshift* you're prompted to set up your backups. Before proceeding any further, make sure you have a suitable backup drive ready – while *Timeshift* can store snapshots on your system

» MINT'S BUILT-IN BACKUP TOOL

Linux Mint ships with a file-based backup tool – *MintBackup* – that offers two separate backups in one app: Personal Data backs up the contents of your **Home** directory, while Software Selection covers the apps installed on your PC through *Software Manager.* You can include apps installed elsewhere in the list by issuing the following command in the terminal before opening *Backup:*

```
$ dpkg --get-selections > package_list.list
```

Type `cat package_list.list` to view the list of packages.

The app is simple to use – when it comes to file backups, everything in your **Home** directory is selected by default except for hidden files, but once you've set your backup destination (as always, this should be a separate physical drive from the one your **Home** folder currently resides on), you can exclude specific files and directories, then choose whether to include specific hidden directories or files – a good idea as many apps store configuration information in these folders. Click Apply and the backup is made.

The big weakness of *MintBackup* is that it can't be set to run automatically on a schedule, which is why we recommend Mint users switch to *Déjà Dup Backups.*

»

QUICK TIP

Redo Rescue saves each partition as a compressed IMG file, along with a text-based REDO file containing details of the drive structure that Redo requires should you need to restore the backup.

partition (as it does by default if you don't choose another location), it's far better to store them separately. This can either be on a dedicated partition on your main drive or – preferably – a dedicated ext4 partition on an external drive for extra security. Use the *Disks* utility or *GParted* to set up an ext4 partition if necessary – *Timeshift* doesn't recognise non-Linux partitions such as FAT, exFAT and NTFS.

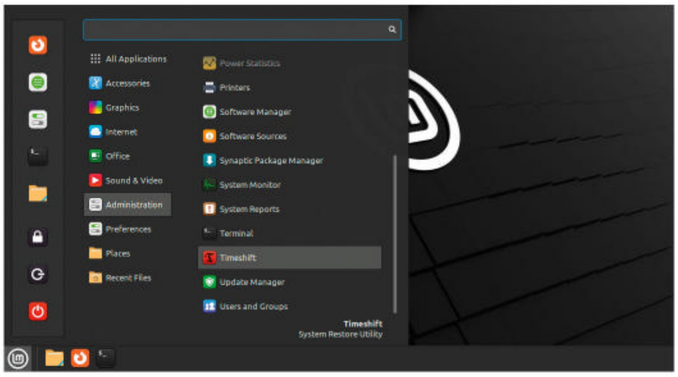
There's a wizard that should open automatically the first time you launch *Timeshift*, or which can be opened via the Wizard button from the main menu. This shows two Snapshot Types – RSYNC or BTRFS – which refer to the underlying command-line tool that *Timeshift* uses. On bog-standard Mint systems, only RSYNC is available, but even where BTRFS is an option, we're sticking with RSYNC for the simple reason that it supports storing snapshots on an external drive.

After selecting RSYNC, click Next and wait for *Timeshift* to estimate the size of your first snapshot, which will be by far the largest backup as subsequent snapshots are incremental, so are much smaller because they only record file changes.

After a short pause, you're prompted to choose your backup location – your system drive is selected by default, so select your backup partition or drive. Click Next and you're prompted to set up a schedule for taking snapshots automatically. By default, the feature is disabled, but you can configure them be taken at Boot (10 minutes after your system has started) as well as set intervals – hourly, daily, weekly and/or monthly. Just tick one or more boxes and select how many of each snapshot to keep. The more backups you create, the more drive space you need, so bear this in mind.

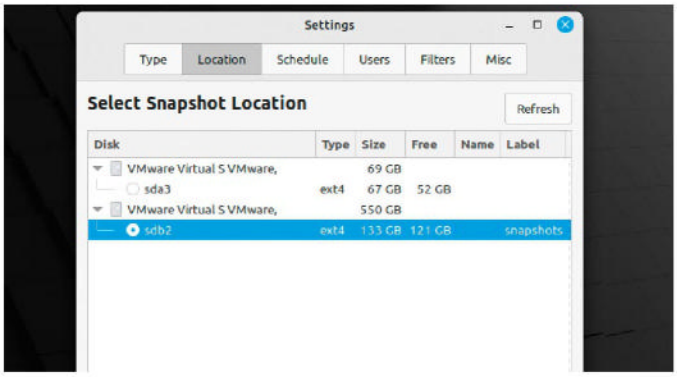
Once you've made your choices, click Next again. You're given the option of including account folders in your snapshot – while it's tempting to suggest selecting Include Only Hidden Files for user folders, to include the **.config** folder where many apps store their preferences, this inflates the size of your snapshots. We recommend using *Timeshift* just to protect system

RESTORE SYSTEM SNAPSHOTS



1 Set up restore

Whether recovering from your current installation, booting from a live rescue environment or restoring a snapshot after reinstalling Linux from scratch, the process is virtually the same. Start by opening *Timeshift* – if you're running from an Ubuntu live environment, you first need to install it (`sudo apt timeshift`) before opening it from the launcher.

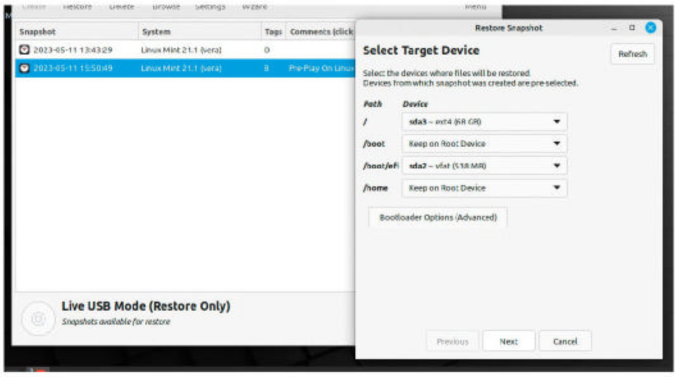


Select Snapshot Location

Disk	Type	Size	Free	Name	Label
VMware Virtual S VMware,		69 GB			
○ sda3	ext4	67 GB	52 GB		
VMware Virtual S VMware,		550 GB			
● sdb2	ext4	133 GB	121 GB		snapshots

2 Select backup to restore

If the *Timeshift* wizard pops up, click Finish to close it, followed by Settings > Location tab. Select the drive containing your snapshots and click OK, which brings your snapshots into view. Identify the one you wish to roll back to (in most cases, this should be the most recent snapshot), then select it before clicking the Restore button.



Snapshot	System	Tags	Comments (click
2023-05-11 13:43:29	Linux Mint 21.1 (vera)	O	
2023-05-11 15:50:49	Linux Mint 21.1 (vera)	B	Pre-Play On Linu

Select Target Device

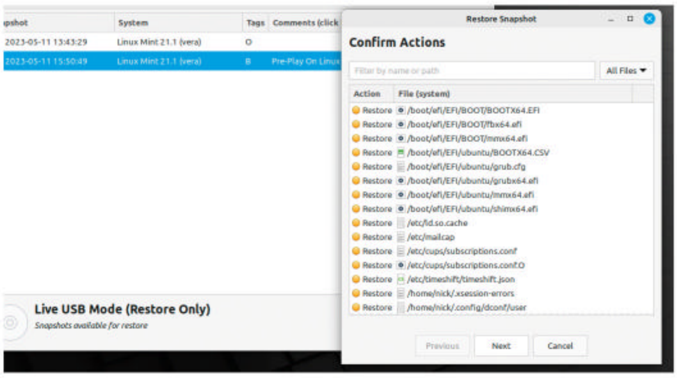
Select the devices where files will be restored.
Devices from which snapshot was created are pre-selected.

Path	Device
/	sda3 – ext4 (68 GB)
/boot	Keep on Root Device
/boot/efi	sda2 – vfat (538 MB)
/home	Keep on Root Device

Live USB Mode (Restore Only)
Snapshots available for restore

3 Select target

Timeshift prompts you to choose which devices to restore your backed up files to – by default, these should already be set to the correct locations, but verify each using its drop-down menu and change if necessary. Click Bootloader Options (Advanced) if you don't want to roll back the *GRUB* settings. Click Next.



pshot	System	Tags	Comments (click
2023-05-11 13:43:29	Linux Mint 21.1 (vera)	O	
2023-05-11 15:50:49	Linux Mint 21.1 (vera)	B	Pre-Play On Linux

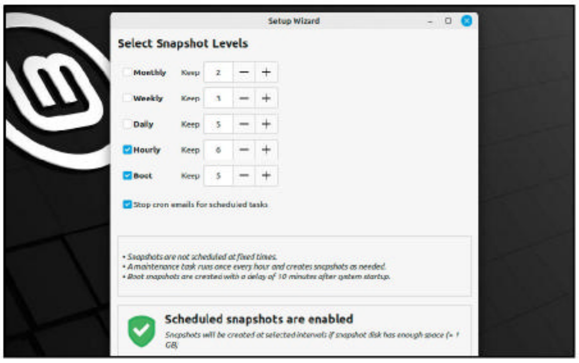
Confirm Actions

Action	File (system)
Restore	/boot/efi/EFI/BOOT/BOOTX64.EFI
Restore	/boot/efi/EFI/BOOT/fbx64.efi
Restore	/boot/efi/EFI/BOOT/mmx64.efi
Restore	/boot/efi/EFI/ubuntu/BOOTX64.CSV
Restore	/boot/efi/EFI/ubuntu/grub.cfg
Restore	/boot/efi/EFI/ubuntu/grubx64.efi
Restore	/boot/efi/EFI/ubuntu/mmx64.efi
Restore	/boot/efi/EFI/ubuntu/shimx64.efi
Restore	/etc/ld.so.cache
Restore	/etc/mailcap
Restore	/etc/cups/subscriptions.conf
Restore	/etc/cups/subscriptions.conf.O
Restore	/etc/timeshift/timeshift.json
Restore	/home/nick/.xsession-errors
Restore	/home/nick/.config/dconf/user

Live USB Mode (Restore Only)
Snapshots available for restore

4 Restore and reboot

Timeshift performs a dry run before providing a list of what files will be restored, created or deleted. You can filter the list or search for specific files to confirm changes you expect to be made. Once happy, click Next and the files are restored. Reboot your system to complete the rollback.



Setup Wizard

Select Snapshot Levels

☐ Monthly Keep 2
☐ Weekly Keep 3
☐ Daily Keep 5
☑ Hourly Keep 6
☑ Boot Keep 5

☑ Stop cron emails for scheduled tasks

- Snapshots are not scheduled at fixed times.
- A maintenance task runs once every hour and creates snapshots as needed.
- Boot snapshots are created with a delay of 10 minutes after system startup.

Scheduled snapshots are enabled

Snapshots will be created at selected intervals if snapshot disk has enough space (> 1 GB)

It's a fine balance choosing how often to create system snapshots – the more you have, the better your restore options are.

files (let *Backups* or *Back In Time* protect your **Home** folder and app configuration files), so leave it set to Exclude All Files before clicking Next to see a summary of settings. Click Finish to return to the main screen.

Take your first snapshot

Click Create to set up your first snapshot. This takes some time, and when it's done the snapshot is listed in the window along with its time and date, system version, tags (one or more letters identifying the type of snapshot, such as O for on-demand and B for boot), and comments. The last one is editable, and enables you to add some descriptive information if you want.

From now on, *Timeshift* automatically takes snapshots according to the schedule you choose, and each one shows up here. You can also manually take snapshots using the Create button – for example, before updating to the next major version of Mint or Ubuntu – which is when the Comments field is particularly useful, enabling you to easily identify specific snapshots going forward.

Should you ever need to restore a system snapshot, the step-by-step guide *(opposite)* reveals what you need to do, from a simple restore within your current Linux installation to recovering a non-booting system with the aid of your Linux live DVD or flash drive.

A fail-safe backup

As things stand, you've backed up your **Home** folder and instigated a snapshot tool to protect system files. But what if you accidentally miss a file or setting? The answer lies in taking a fail-safe drive image – a complete byte-by-byte backup of your system drive.

While it's possible to take a drive image using the *Disks* utility or using `dd` via the terminal, this can be incredibly slow (and in the case of *Disks*, only produces uncompressed images). A tool called *Redo Rescue* (**http://redorescue.com**) provides you with an emergency boot disk that can be used to both create and restore disk and partition images.

After booting from the disk, *Redo Rescue* provides a basic desktop interface, but launches straight into the *Redo Rescue* tool. Start by clicking Backup, select your boot drive and click Next. By default, all partitions on the drive (including bootable partitions) are selected, so click Next again to select the destination drive – this can be a local disk, network folder or even NFS/SSH/FTP-connected storage – and the folder in which the image will be stored as a single file.

You're then prompted to give the backup a suitable name and description before clicking Next to start the backup process. Once complete, click Verify at the main screen to ensure the file isn't corrupt, and then – if you're being thorough – take a second fail-safe drive image that you store on another drive.

Should the drive containing your Linux installation subsequently fail, you can install a replacement, then boot from your *Redo Rescue* emergency media. Click Restore, then browse to the folder containing your drive image to open the **.redo** file. Finally, select your newly installed drive, stay on the Full System Recovery tab (Restore Data Only is an option if your old drive is still functioning and you just want to restore individual partitions), then click Next followed by Yes, I'm Sure!

Once done, reboot into your restored Linux installation. If the drive image is older than your most recent system snapshot and/or file-based backup, you can then use the restore tools in *Timeshift* and *Backups* to bring your system back up to date. LXF

QUICK TIP

To make changes to your Timeshift configuration, either click Wizard to run through the setup wizard again, or Settings to reconfigure specific areas of your configuration, including areas not covered by the wizard, such as include/exclude filters.

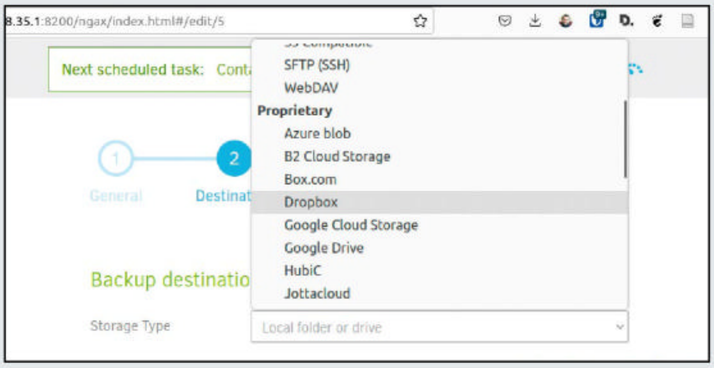
» BACK UP SECURELY TO THE CLOUD

It's always a good idea to keep at least one copy of your data stored in a separate location from your PC. Cloud storage offers the most convenient way to do this, but there are question marks over security. One way to get around this is to wrap your backup files in another layer of encryption before uploading them, and the simplest way to do that is with a tool called *Duplicati* (**www.duplicati.com**).

Duplicati is supplied as a DEB file, so once downloaded, double-click this to install it via *Software Manager* with its dependencies (mono being the main prerequisite). Open *Duplicati* and it redirects you to a browser window and the main web-based interface.

From here it's relatively straightforward. Click Add Backup to work your way through the wizard – you start by entering a passphrase with which to encrypt your backup files (make a note of this) before choosing where to store your backup files. *Duplicati* supports local storage, various network protocols and a range of proprietary cloud storage providers, including Dropbox, Mega, Google Drive and Microsoft OneDrive. Choose what to back up, set a schedule and select your retention options, then leave *Duplicati* to do the rest.

For more on *Duplicati,* check out our tutorial from **LXF263** (available to subscribers at **www.linuxformat.com/archives?issue=263**).



8.35.1:8200/ngax/index.html#/edit/5

Next scheduled task: Cont

1 General — 2 Destinat

Backup destinatio

Storage Type: Local folder or drive

Duplicati's big selling point is that you can store your backups securely on a variety of different cloud-based platforms.

PL/I

Credit: www.iron-spring.com

PL/I – the multi-faceted language

Mike Bedford discovers the mysteries of PL/I, the language that IBM hoped would replace both FORTRAN and COBOL.

OUR EXPERT

Mike Bedford is never one to turn down a new challenge. He discovered that trying PL/I is an interesting experience, but there are undoubtedly better languages today for real-world applications.

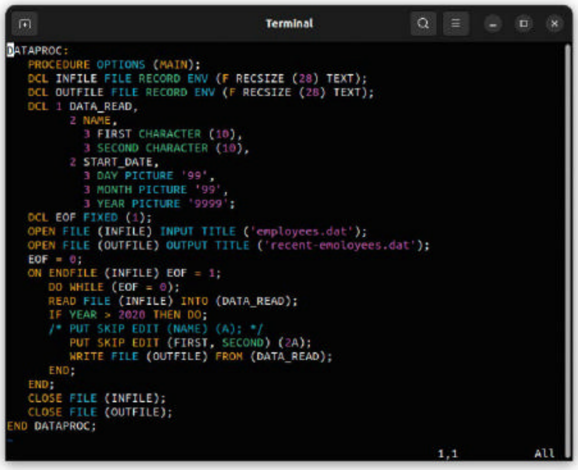
QUICK TIP

Despite our describing some of PL/I's features as COBOL-like, there's one very noticeable difference between the two languages. A bizarre concept in the design of COBOL, which PL/I did not mimic, was that statements should look like English language sentences.

The names of programming languages vary from the banal to the accurately descriptive, via the totally meaningless or, in the case of this month's subject, the downright inaccurate. The language in question is PL/I. The letter I is the Roman numeral for the figure one, which explains why it's occasionally, incorrectly, shown as PL/1. It stands for Programming Language One, but it certainly wasn't. In fact, two of the languages we've previously covered in this series predate it, as do several others.

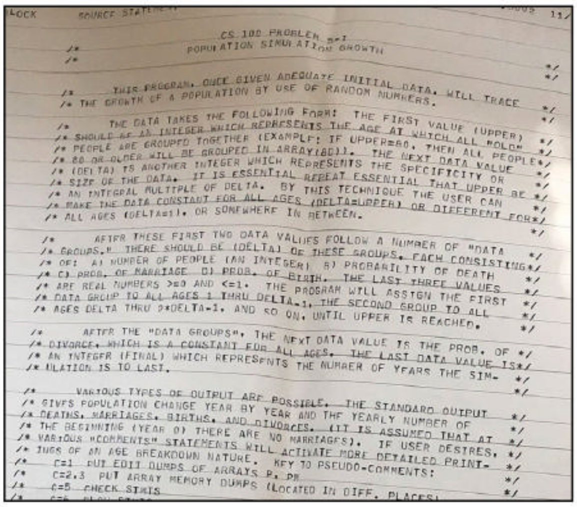
To make it into our list of classic languages, a language has to have been around for a while, to put it politely, and PL/I certainly ticks that box, having been launched in 1964. It was designed by IBM for use on its System/360 mainframes, and was first used at its Hursley Laboratories in the UK as part of the 360 development programme.

To set the scene, let's consider the main languages that had previously been promoted by IBM in the early '60s. FORTRAN was used for scientific applications, and COBOL was used for business applications. While so much more basic, FORTRAN offered the same types of instructions provided by today's common languages, but COBOL was very different. Because of its emphasis on data handling for commercial jobs, it became the first language to allow hierarchical data structures to be defined and subsequently manipulated. The aim of PL/I was to merge the features of these two languages and thereby provide a solution that would address the needs of both scientific or technical users and business users.



```
DATAPROC:
   PROCEDURE OPTIONS (MAIN);
   DCL INFILE FILE RECORD ENV (F RECSIZE (28) TEXT);
   DCL OUTFILE FILE RECORD ENV (F RECSIZE (28) TEXT);
   DCL 1 DATA_READ,
         2 NAME,
           3 FIRST CHARACTER (10),
           3 SECOND CHARACTER (10),
         2 START_DATE,
           3 DAY PICTURE '99',
           3 MONTH PICTURE '99',
           3 YEAR PICTURE '9999';
   DCL EOF FIXED (1);
   OPEN FILE (INFILE) INPUT TITLE ('employees.dat');
   OPEN FILE (OUTFILE) OUTPUT TITLE ('recent-emoloyees.dat');
   EOF = 0;
   ON ENDFILE (INFILE) EOF = 1;
      DO WHILE (EOF = 0);
      READ FILE (INFILE) INTO (DATA_READ);
      IF YEAR > 2020 THEN DO;
      /* PUT SKIP EDIT (NAME) (A); */
         PUT SKIP EDIT (FIRST, SECOND) (2A);
         WRITE FILE (OUTFILE) FROM (DATA_READ);
      END;
   END;
   CLOSE FILE (INFILE);
   CLOSE FILE (OUTFILE);
END DATAPROC;
                                                            1,1           All
```

A PL/I syntax highlighting add-on is available for the Vi text editor – especially useful with an unfamiliar language.



```
LOCK      SOURCE STATEMENT                                               11/
                          CS 100 PROBLEM 5-1
     /*              POPULATION SIMULATION GROWTH
     /*                                                                    */
                                                                           */
     /*        THIS PROGRAM, ONCE GIVEN ADEQUATE INITIAL DATA, WILL TRACE  */
     /* THE GROWTH OF A POPULATION BY USE OF RANDOM NUMBERS.               */
     /*        THE DATA TAKES THE FOLLOWING FORM:  THE FIRST VALUE (UPPER) */
     /* SHOULD BE AN INTEGER WHICH REPRESENTS THE AGE AT WHICH ALL "OLD"   */
     /* PEOPLE ARE GROUPED TOGETHER (EXAMPLE: IF UPPER=80, THEN ALL PEOPLE */
     /* 80 OR OLDER WILL BE GROUPED IN ARRAY(80)).  THE NEXT DATA VALUE    */
     /* (DELTA) IS ANOTHER INTEGER WHICH REPRESENTS THE SPECIFICITY OR     */
     /* SIZE OF THE DATA.  IT IS ESSENTIAL REPEAT ESSENTIAL THAT UPPER BE  */
     /* AN INTEGRAL MULTIPLE OF DELTA.  BY THIS TECHNIQUE THE USER CAN     */
     /* MAKE THE DATA CONSTANT FOR ALL AGES (DELTA=UPPER) OR DIFFERENT FOR */
     /* ALL AGES (DELTA=1), OR SOMEWHERE IN BETWEEN.                       */
     /*        AFTER THESE FIRST TWO DATA VALUES FOLLOW A NUMBER OF "DATA  */
     /* GROUPS."  THERE SHOULD BE (DELTA) OF THESE GROUPS, EACH CONSISTING */
     /* OF:  A) NUMBER OF PEOPLE (AN INTEGER)  B) PROBABILITY OF DEATH     */
     /* C) PROB. OF MARRIAGE  D) PROB. OF BIRTH.  THE LAST THREE VALUES    */
     /* ARE REAL NUMBERS >=0 AND <=1.  THE PROGRAM WILL ASSIGN THE FIRST   */
     /* DATA GROUP TO ALL AGES 1 THRU DELTA-1, THE SECOND GROUP TO ALL     */
     /* AGES DELTA THRU 2*DELTA-1, AND SO ON, UNTIL UPPER IS REACHED.      */
     /*        AFTER THE "DATA GROUPS", THE NEXT DATA VALUE IS THE PROB. OF */
     /* DIVORCE, WHICH IS A CONSTANT FOR ALL AGES.  THE LAST DATA VALUE IS */
     /* AN INTEGER (FINAL) WHICH REPRESENTS THE NUMBER OF YEARS THE SIM-   */
     /* ULATION IS TO LAST.                                                */
     /*        VARIOUS TYPES OF OUTPUT ARE POSSIBLE.  THE STANDARD OUTPUT  */
     /* GIVES POPULATION CHANGE YEAR BY YEAR AND THE YEARLY NUMBER OF      */
     /* DEATHS, MARRIAGES, BIRTHS, AND DIVORCES.  (IT IS ASSUMED THAT AT   */
     /* THE BEGINNING (YEAR 0) THERE ARE NO MARRIAGES).  IF USER DESIRES,  */
     /* VARIOUS "COMMENTS" STATEMENTS WILL ACTIVATE MORE DETAILED PRINT-   */
     /* INGS OF AN AGE BREAKDOWN NATURE.  KEY TO PSEUDO-COMMENTS:          */
     /*    C=1  PUT EDIT DUMPS OF ARRAYS P, PR                             */
     /*    C=2,3  PUT ARRAY MEMORY DUMPS (LOCATED IN DIFF. PLACES)         */
     /*    C=5  CHECK STATS                                                */
```

Multiple iterations, even of simple programs, are to be expected with a new language. But at least you won't be using up whole forests in fan-fold listings, as early PL/I programmers would have done.

CREDIT: Jonathan Schilling, CC BY-SA 4.0, https://bit.ly/lxf304cs100

But it went further. While virtually all the earlier languages had little in the way of features to support block-structured programming, ALGOL, our featured language in **LXF302**, and which appeared in 1960, did adhere to this important programming paradigm, so paved the way to nearly all today's popular languages. ALGOL was never really promoted by IBM to any great extent but, even so, its block-structured approach also influenced the design of PL/I. We could almost think of PL/I, therefore, as a direct descendent of FORTRAN, COBOL and ALGOL, even though it didn't have a great deal of success in replacing any of these languages.

Do it yourself

You'll probably want to try your hand at churning out and running some PL/I code, but the options aren't as plentiful as they were for our previous classic languages. We couldn't find any online resources for running PL/I, so we're stuck with compilers. Sadly, what appears to be the only open source offering hasn't been updated for quite some time. So, instead, we're recommending the freely available *Iron Spring* implementation, even though it's closed source.

You can find the latest version of the compiler for Linux at **www.iron-spring.com**, and documentation is also provided in the distribution. The documentation includes **readme_linux.html**, which describes how to install and run the compiler, and **prog_guide.html**, which details, among other things, the differences – mainly in the area of input and output – between Iron Spring PL/I and the version it's based on, namely IBM PL/I for MVS and VM.

There's no complete language reference manual yet, but you can find IBM's original manual at **http://teampli.net/SC26-3114-01.pdf**. Iron Spring PL/I isn't distributed as an executable, but compiling it from the source code provided was straightforward using the supplied **makefile**. A few sample PL/I programs also form part of the distribution, and there are a few more on the *Iron Spring* website, so you can put the compiler through its paces before starting to hack your own code. You'll also find some PL/I code at Rosetta Code, but we recommend trying *Iron Spring's* samples as a very first step. You'll find them in a folder called **Samples**, and if you issue the command `make –f Makefile.Linux` from that folder, all the samples will be compiled and linked, after which you can execute them from the command-line prompt.

If you want to try editing one of them as a coding exercise, edit the source code, delete the other files with the same name but different extensions, most commonly .lst and .obj, and one with no extension, which is the executable, although this depends on the complier options. Then re-issue the `make` command, which will now compile just your edited code. And when you progress to writing your own PL/I code from scratch, you'll need to create your own makefiles – see the samples provided.

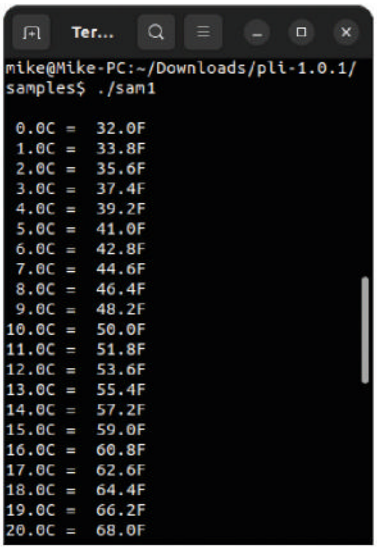
The other practical advice we can offer concerns syntax highlighting. Since you won't be using an IDE, the only interactive tool you'll be using for creating and subsequently editing your PL/I code is a text editor. You could use any of the many FOSS editors available, although most don't have syntax highlighting for PL/I, at least not as standard. We did discover that a PL/I syntax highlighting add-on is available for the *Vim* editor, though, and you can find it at **www.vim.org/scripts/script.php?script_id=4329**. This is highly recommended, especially as PL/I will probably be unfamiliar to you. You can see it in action in our screenshot *(opposite)*, but you'll also notice that there appear to be a few quirks, so be wary of trusting it explicitly. While we haven't fully got up to speed with which colours are used for what, the screenshot, which shows our second example program being edited, seems to have a few inconsistencies. For example, one of our variable names, **FIRST**, is coloured differently from most from the others – it's white instead of green. We assume this is a consequence of our inadvertently using a keyword as a variable name, which is legal because PL/I doesn't have reserved words, even though it's not really recommended. We can't help but feel that had the syntax highlighting been cleverer, say by considering the context, these spurious examples would have been coloured green as variable names.

Replacing FORTRAN

Since, in one guise, PL/I was seen as a potential replacement for FORTRAN, we'll start by looking at a simple program for arithmetic manipulation. The code, which appears below, displays a table of Celsius temperatures together with their Fahrenheit equivalents, for Celsius temperatures starting at 0°C, and increasing in one degree intervals, until we get to a Fahrenheit value of over 100°F.

```
CTOF:
  PROCEDURE OPTIONS (MAIN);
  DCL (F, C) FLOAT;
  DO  C = 0 to 100 UNTIL (F>100);
    F = C * 1.8 + 32;
    PUT SKIP EDIT (C, ‘C = ’, F, ‘F’) (F(4,1), A, F(5,1), A);
    END;
END CTOF;
```

Although we've introduced it as a FORTRAN-type application, we've deliberately chosen to use something that is decidedly non-FORTRAN-like, namely a **DO … UNTIL** loop. In contrast to FORTRAN, which in its original guise had the basic **DO** loop (such as `DO 10 N = 1, 100`) as its only block-structured instruction, PL/I took inspiration from ALGOL by offering several ways to structure code in blocks. In fact, the whole of the code is defined as a block – specifically the main procedure **CTOF**, which is bracketed by `CTOF:` and `END CTOF` – and had it been a more complicated program, it would have been



```
mike@Mike-PC:~/Downloads/pli-1.0.1/
samples$ ./sam1

 0.0C =  32.0F
 1.0C =  33.8F
 2.0C =  35.6F
 3.0C =  37.4F
 4.0C =  39.2F
 5.0C =  41.0F
 6.0C =  42.8F
 7.0C =  44.6F
 8.0C =  46.4F
 9.0C =  48.2F
10.0C =  50.0F
11.0C =  51.8F
12.0C =  53.6F
13.0C =  55.4F
14.0C =  57.2F
15.0C =  59.0F
16.0C =  60.8F
17.0C =  62.6F
18.0C =  64.4F
19.0C =  66.2F
20.0C =  68.0F
```

Forget about fancy IDEs in your exploration of PL/I. You'll be editing your code in a plain text editor, and you'll be running it at the terminal prompt.

BACK IN THE USSR

PL/I might not have enjoyed a great deal of success in the West, but things were very different behind the Iron Curtain in the early '70s. It appears that PL/I was used extensively in the Soviet Union, and other countries under its sphere of influence, where it came a lot closer to doing what IBM had hoped for it, namely to supplant both FORTRAN and COBOL. However, like so many things in the former USSR, definitive information is hard to come by.

IBM's development of the general-purpose language PL/I was very much tied up with its development of the System/360, which was conceived as a general-purpose computer. This was in marked contrast to its previous machines, which were aimed either at scientific or commercial users. Given that the Soviet ES EVM series computers were clones of System/360 mainframes – reportedly produced by reverse engineering – it would surely have been quite natural for PL/I to be key to the ES EVM programme. And the limited documentation available does, indeed, show PL/I as being offered with ES EVM machines.

What we don't know, however, is whether the Soviet version of PL/I was identical to that used in the West, or whether it was a nationalised version, as was the case with some Soviet versions of ALGOL. Or, to put it another way, would the second line of our second sample program have been initially as meaningless to Russian-speaking programmers as something not too dissimilar to ПРОЦЕДУРЫ ВАРИАНТЫ (ОСНОВНЫЕ) would be to most of us?

As evidence of its suitability for system programming, PL/I was used to write the Multics time-sharing operating system for the GE-645, a successor to this General Electric 400 series machine.

CREDIT: Marcin Wichary, CC BY 2.0, www.flickr.com/photos/mwichary/30756156060/

QUICK TIP

It's notable that some versions of PL/I featured recursion, but not all later versions. Given that each time a recursive function was called, the values of all its local variables were placed on the stack, this would rapidly eat up a valuable commodity in those early days: memory.

advisable to split it up into other procedures that would be called from the main procedure.

The code should mostly make sense with the possible exception of the **PUT** statement that writes output to the console, which back in the day would probably have been a hard copy teletype terminal, but in our case us the terminal window. The contents of the first set of parentheses in the **PUT** statement lists the items to be displayed, specifically a mixture of the variables **C** and **F** plus a couple of text strings. And the code in the second set of parentheses defines the display format so, for example, `F(4,1)` defines that the number should be a four-character floating point number (that number including the decimal point), of which one of the digits is after the decimal point.

Replacing COBOL

Next up we'll delve into a program that performs some simple file handing and data manipulation, which is typical of a COBOL application and appears below:

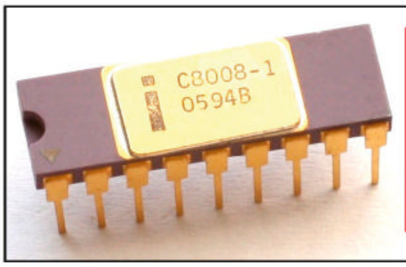
```
DATAPROC:
  PROCEDURE OPTIONS (MAIN);
  DCL INFILE FILE RECORD ENV (F RECSIZE (28)
TEXT);
  DCL OUTFILE FILE RECORD ENV (F RECSIZE (28)
TEXT);
  DCL 1 DATA_READ,
      2 NAME,
        3 FIRST CHARACTER (10),
        3 SECOND CHARACTER (10),
      2 START_DATE,
        3 DAY PICTURE '99',
        3 MONTH PICTURE '99',
        3 YEAR PICTURE '9999';
  DCL EOF FIXED (1);
  OPEN FILE (INFILE) INPUT TITLE ('employees.dat');
  OPEN FILE (OUTFILE) OUTPUT TITLE ('recent-
emoloyees.dat');
  EOF = 0;
  ON ENDFILE (INFILE) EOF = 1;
    DO WHILE (EOF = 0);
    READ FILE (INFILE) INTO (DATA_READ);
    IF YEAR > 2020 THEN DO;
    /* PUT SKIP EDIT (NAME) (A); */
      PUT SKIP EDIT (FIRST, SECOND) (2A);
      WRITE FILE (OUTFILE) FROM (DATA_READ);
    END;
  END;
  CLOSE FILE (INFILE);
  CLOSE FILE (OUTFILE);
END DATAPROC;
```

The purpose of this code is to read records from a file containing data relating to a company's employees, and write the same details to another file for those members of staff who started their employment after 31st December 2020, while also listing the names of those same employees to the console. The data as read from the file is structured hierarchically, with the complete record (**RECORD_READ**) being defined in the line starting `DCL 1`. `DCL` is short for `DECLARE`, and you can use `DECLARE` in full if you prefer, and the screenshot *(page 60)* shows that **RECORD_READ** is a first-level variable. **RECORD_READ** is split into two second-level variables, **NAME** and **START_DATE**; **NAME** is divided into two third-level variables, **FIRST** and **SECOND**; and **START_DATE** is also divided into third-level variables, namely **DAY**, **MONTH** and **YEAR**. Of course, had this been a real-world application, the record would have included much more information about the employees, but we've omitted that so as not to waste space with code we're not going to use. Nevertheless, this is enough to illustrate an important PL/I concept, which it had inherited from COBOL, namely its support for hierarchical data structures.

This hierarchical approach allows variables at any level to be manipulated. So, for example, the whole of the record (**RECORD_READ**), which is at level one, is read from and written to file in single instructions, a selection is made based on the third-level variable **YEAR**, and names are written to the console using the third-level variables **FIRST** and **SECOND**. Actually, since it's strictly correct, it should have been possible to write the statement that refers to **FIRST** and **SECOND** to refer, instead, to the single second-level variable **NAME**, as shown in the statement that is commented out, by being enclosed in a `/* … */`. However, this didn't work, and it was confirmed by *Iron Spring* as a glitch in the compiler which will be fixed in the next release, hopefully by the time you read this.

Having explained the concept of hierarchical data structures, while the syntax will probably be different from that of any other languages you might be familiar with, we suspect that you'll have little difficulty figuring out how it works, so we'll pass little comment on it. And we will gloss over the statements that start with `DCL INFILE` and `DCL OUTFILE` – after all, it's not our intention here to turn you into a fully-fledged PL/I programmer. A comment on the `ON ENDFILE` statement is appropriate, though. You can probably guess what it does but, reportedly, "PL/1 was the first programming language that provides correct handing of such a common problem as detection of the end of the sequential file while reading."

To use this code, you need to create an ASCII file called **employees.dat**. This should be a 28-character fixed length file, and each item of data in the file should be a fixed length, as defined in the PL/I code. So, the first and second names should have trailing spaces to make them up to the required 10 characters, and any single-digit days or months should have a leading zero. You could create a different file, although it should still be called **employees.dat**, but the following is our test



PL/I predated the microprocessor by seven years but that didn't stop it appearing on microchips, including the Intel 8008, thanks to PL/M.

file, and the records that the code will write to the file **recent-employees.dat** will be **FRED BEDFORD** and **HARRY HILL**:

```
FRED      BEDFORD   01012023
GEORGE    SMITH     10102019
HARRY     HILL      08082022
JENNY     MARSHAL   05052018
```

Note that there shouldn't be an **LF** terminator on the final line of the file, otherwise the PL/I program will read an empty line at the end, and this will cause some unexpected behaviour. Some text editors, including *Vim* and the *GUI Text Editor* that forms part of the Ubuntu distribution, do add an **LF** on the last line, but we discovered that *Brackets* doesn't, so that would be a good one to use.

» PL/I'S IMMEDIATE FAMILY

Most of us have seen 'family trees' that show how programming languages have been influenced by their predecessors and, in turn, themselves influenced their successors. However, rarely do these diagrams show today's languages as being in PL/I's direct line of descent. In fact, it seems it was something of a dead end. However, it did spawn several languages in the short term, most of them having names starting PL/, to emphasise their family resemblance.

To be honest, many of PL/I's descendents were really dialects of the original language, and here we could mention PL/M, which was developed by Intel for use on microprocessors. PL/S was a bit different in being further tailored for system programming and, as an extension to PL/I, it allowed assembly code to be included inline.

PL/C, Programming Language Cornell, was developed at Cornell University. It was designed as an educational language, so in the early '70s, it would have been an alternative to BASIC, despite the two languages being very different. In reality, despite being referred to as a dialect, PL/C was little changed from PL/I. However, the implementation was novel in that students never encountered compilation errors, which was important when it might have taken hours to receive the listing from a compilation. To achieve that, it attempted to correct syntax errors and, where it wasn't possible, it converted the offending statements to output to help in debugging.

PL/I as a system language

Less well known, perhaps, than PL/I's use as both a scientific/technical/mathematical language and for commercial, business-oriented applications, PL/I was also designed for system programming. As such, it can be likened to C, plus several more modern languages, in straddling the divide between the domains of system and application programming.

Prior to PL/I, assembler had been used pretty much exclusively for writing operating systems and compilers. Partly this was due to efficiency, and we admit that the design of the language alone wouldn't address that directly, because efficiency is very much tied up with the performance of the compiler. However, language features are also highly important. After all, if you're coding in assembler, you can do pretty much anything, whereas this wasn't the case with many of the early high-level languages. Having never written any system code, this author couldn't say first-hand what features are required by the system programmer. However, we did come across the following statement from an unnamed person who had been involved in writing PL/I compilers, which provides just a little insight: "PL/1 was the first general-purpose language with usable string data type. It also has variable-length bit strings." Needless to say, it also provided a means of working with those new types of data, for example by providing the bitwise operators **OR**, **AND**, **XOR** and **NOT**, something that the original versions of FORTRAN and COBOL didn't offer. Or, to be more accurate, you can do this in any language, of course, as all practical languages are universal, but only where bitwise operations are provided natively is this efficient.

Probably the best way of seeing something of PL/I's capabilities in the system programming arena, though, is to look at Multics, where it pioneered the use of a high-level language for writing operating systems. Multics was one of the first multi-user, time-sharing operating systems. It's considered to have influenced all modern operating systems. Indeed, it's claimed that its most lasting legacy on the computer industry was to inspire the creation of UNIX, and we all know where that lead. It also seems to have been a commercial success, since it ran at a total of 85 sites. That might not sound like a lot today, but this was in an era when computers were vastly expensive and therefore few and far between. The Multics project started in 1964, with the initial release in 1969. It was a joint project involving MIT, General Electric (GE), and Bell Labs. It was developed on the GE-645 mainframe computer, which was specially designed for it. General Electric is barely remembered as a computer manufacturer today. However, its business, including Multics, was taken over by Honeywell in 1970, which later became the French computer company Bull, and is now part of the multinational IT company ATOS. LXF

While not a major success in the West, PL/I was used extensively in the USSR, on machines like this ES EVM 1035 mainframe.

QUICK TIP

PL/I is certainly not dead and gone – reportedly, the language is still used for new developments, commonly for systems programming, primarily on IBM z/Architecture mainframes.

CREDIT: Konstantin Lanzet, https://commons.wikimedia.org/wiki/File:KL_Intel_C8008-1.jpg
CREDIT: Deutsche Fotothek, CC BY-SA 3.0 DE, https://commons.wikimedia.org/wiki/File:Fotothek_df_n-11_0000385.jpg

BACK ISSUES » MISSED ONE?

ISSUE 303
July 2023

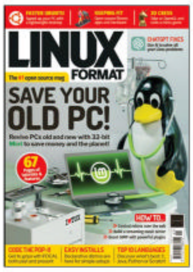
Product code:
LXFDB0303

In the magazine
Trim down your Ubuntu installation for a stripped-back, super-slick and ultra-fast Linux experience. Plus, discover which beginner programming language is the best, read all about OpenSUSE's ALP distro, and find out why *Podman* might be better than *Docker.* And don't forget to try our tutorials on terminal-based to-do lists, searching file contents, making mazes in C, and loads more.

ISSUE 302
June 2023

Product code:
LXFDB0302

In the magazine
Master hacking with Kali Linux, discover the artful ways of *Stable Diffusion,* try a taste of the anything-but-plain Vanilla OS and find out what's new in the online office world. Plus, pick up a whole bunch of new skills thanks to tutorials on speeding up text-based tasks, rescuing data from floppies, repairing images in *GIMP* and much more, and learn which is the hottest firewall distro in our *Roundup.*



ISSUE 301
May 2023

Product code:
LXFDB0301

In the magazine
Save your old PC by installing Linux Mint on ageing hardware. Also in this issue, we explore the programming language leaderboard, look at keeping fit the open source way, feature a *Roundup* of lightweight desktops and spoil you with tutorials on everything from image processing and music streaming to programming a 3D chess game and controlling your DIY robot.

ISSUE 300
April 2023

Product code:
LXFDB0300

In the magazine
The 300th issue celebrates the very best open source apps and revisits the birth of the free software movement. We also take an in-depth look at the ups and downs of *Firefox* and enjoy some retro distro reviews from the early days of the magazine. There's a *Roundup* of current key distros, too, plus tutorials on Debian Hurd, *GIMP,* Mastodon and coding your own *Pong* game, to name just a few.

ISSUE 299
March 2023

Product code:
LXFDB0299

In the magazine
Free your phone and power it up with Linux and a host of open source apps. Plus, discover how to deck out your Steam Deck to get even more from Valve's gaming device, and find out what immutable OSes can offer the future of Linux. You'll also find a *Roundup* of packaging formats, reviews of new distros, tutorials on building a robot, batch converting images, algorithmic art and more.

ISSUE 298
February 2023

Product code:
LXFDB0298

In the magazine
Dig in to our New Year smorgasbord of hot Pi projects, from Pico devices running MicroPython to full-Pi setups. Plus, discover how you can contribute to open source projects without being a coder. We also have a *Roundup* of Matrix messaging clients, and a torrent of tutorials on everything from tackling tables to planning a novel, polishing video footage, using *GIMP* and lots more besides.

To order, visit www.magazinesdirect.com

Select **Single Issues** from the tab menu, then select **Linux Format**.

Or call the back issues hotline on **0330 333 1113**
or **+44 (0)330 333 1113** for overseas orders.

Quote the product code shown above and have your credit or debit card details ready.

USA? EU? THE MOON*?

UK readers turn to p16

SUBSCRIBE!

Don't wait for the latest issue to reach your local store – subscribe today and let *Linux Format* fly straight to you. Faster, cheaper and with DRM-free archive access!

FASTER SEARCH Turbocharge your desktop search tool | SUSE ALP Get started with the next-gen SUSE distro | GET CODING We test the most popular languages

LINUX FORMAT

The #1 open source mag

Enhance your Pi cameras

65 pages of tutorials & features

LIGHTSPEED UBUNTU

Take the world's favourite distro to the next level: stripped back, super-slick and ultra-fast!

HOW TO...

» Ride the Microsoft Azure Linux cloud
» Write technical papers with AsciiDoc
» Get to grips with the classic LISP

MUSIC TRACKERS Relive your Amiga days with modern updates | BE A PODMAN Containers made easy thanks to Red Hat | PROGRESS BARS Slick rich-text terminal dialogs using Python

3 GREAT WAYS TO SUBSCRIBE

Print, digital-only, and print+digital bundles!

» USA
From $132
For 13 issues

» REST OF THE WORLD
From $132
For 13 issues

» EUROPE
From €100
For 13 issues

IT'S EASY TO SUBSCRIBE!

Visit **www.magazinesdirect.com/linux-format**

Call **+44 0330 333 1113**

Lines open Monday-Friday, 9am-5pm, UK time.

*We don't actually deliver to the Moon. Yet.

SWAY

Credit: https://github.com/swaywm/sway/

How to test out new desktops the easy way

Never accused of being fast and smooth, **Mats Tage Axelsson** explains how to try your next desktop and make the switch faster and smoother.

OUR EXPERT

Mats Tage Axelsson keeps finding new ways of using Linux for everyday tasks. His desktop stays the same, despite trying – making the laptop dance is still some way off.

We all have a desktop on our computers and they have caused countless flame wars over the years. Barring hardcore hackers who live in a terminal shell, most people want a nice graphical front-end to look at while they work or play.

The two most common choices are Gnome and KDE, which work splendidly for most situations. In spite of this, there's merit in trying other options. Apart from looks, you also have resource usage to consider. If your system is a Raspberry Pi or similar, you may want a more slimline binary.

You can change the look of your desktop on a daily basis, but you may have more pressing demands on your time. It is still interesting to get the desktop to look exactly the way you want, though, so let's see how we can get you desktop surfing more easily…

Desktop hopping

For low-spec systems, you may need a slimmer solution. Common choices are *i3, Awesome* and *dwm.* These are slim and tiling, meaning they occupy the entire screen as default. The standard paradigm is to have applications floating and spread all over the screen. In a tiling window manager, you have a second window on the other half of the screen.

There are other strategies, however, where the main window is always large and the others stay small. These strategies are best to learn by using them,seeing how they fill up your screen. Your situation is unique to you, though, so getting this right takes some exploring.

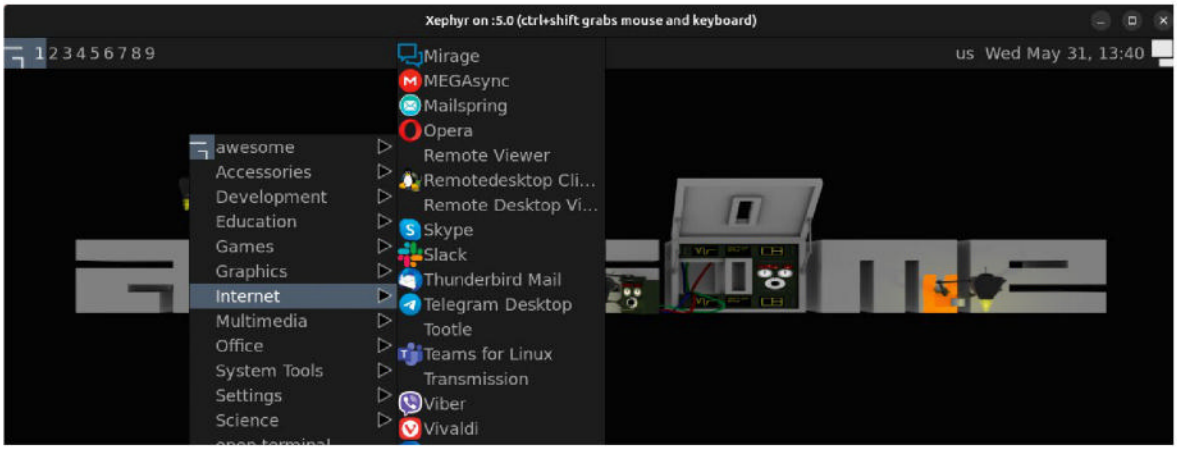
As mentioned, this is even more tedious when you can't try it out on your current desktop. This is where we can bring you hope through our tried-and-tested method – stop hopping and take small steps instead, so you can later take the leap with confidence.

Testing times

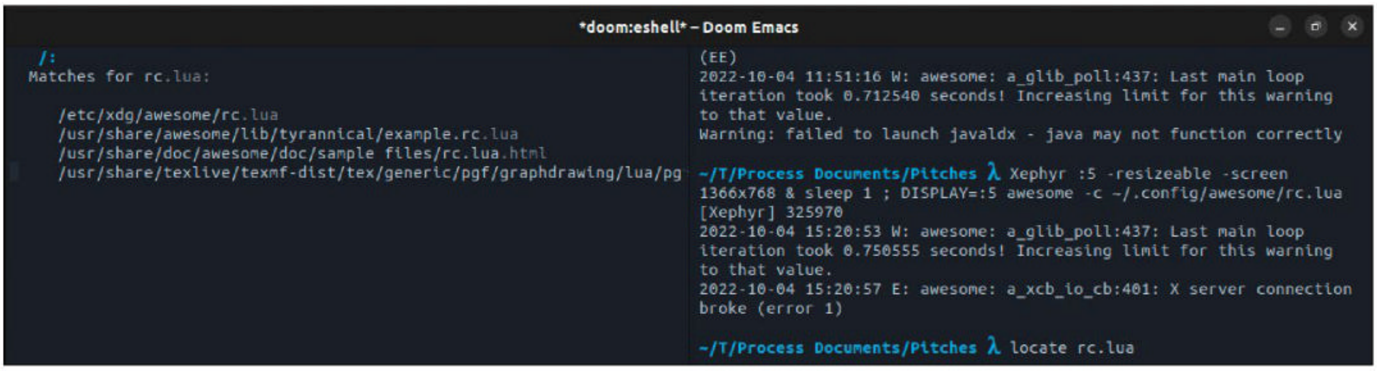
Once you realise that there may be better desktop solutions available, you hit the next obstacle: the learning curve. Even though the concepts are easy to learn, it is hard to get used to new keybindings. You will stop using the mouse for many tasks, if you so choose, and learn to keep some order on your desktop.

All this means you will have a period when you struggle with new ideas and ways of starting your applications. Do you use menus to start applications? You may want to start typing the name instead. Getting used to these things takes some practice, so how do you stay active and productive while you learn?

You could switch to a new window manager and suffer through the bumps. You can also log in, practise, log out and return to old habits. Sometimes you may need to reload because Gnome or KDE start some daemons while slimmer desktops don't. Regarding odd effects from services not running, one example is having the keyring locked – you can add these yourself as you discover the quirks of the various options.



Even the default setting for Awesome has neat features, and it shows you all its apps in a drop-down menu.



doom:eshell – Doom Emacs

```
  /:
Matches for rc.lua:

    /etc/xdg/awesome/rc.lua
    /usr/share/awesome/lib/tyrannical/example.rc.lua
    /usr/share/doc/awesome/doc/sample files/rc.lua.html
    /usr/share/texlive/texmf-dist/tex/generic/pgf/graphdrawing/lua/pg
```

```
(EE)
2022-10-04 11:51:16 W: awesome: a_glib_poll:437: Last main loop
iteration took 0.712540 seconds! Increasing limit for this warning
to that value.
Warning: failed to launch javaldx - java may not function correctly

~/T/Process Documents/Pitches λ Xephyr :5 -resizeable -screen
1366x768 & sleep 1 ; DISPLAY=:5 awesome -c ~/.config/awesome/rc.lua
[Xephyr] 325970
2022-10-04 15:20:53 W: awesome: a_glib_poll:437: Last main loop
iteration took 0.750555 seconds! Increasing limit for this warning
to that value.
2022-10-04 15:20:57 E: awesome: a_xcb_io_cb:401: X server connection
broke (error 1)

~/T/Process Documents/Pitches λ locate rc.lua
```

Using Xephyr and Emacs together gives you a powerful environment to troubleshoot your configuration without breaking your current setup.

A better solution is to open your new window manager while you are using your current one and enter nesting. You can start the new desktop as an application to try it out. This has some drawbacks, but it is great for practice and "ricing" attempts.

VM vs Xephyr

You can do this using a virtual machine, but it causes a lot of load and takes up disk space in spades. *Docker* is another solution, which is faster than virtual machines but less direct than nesting. Both of these options build many more binaries to achieve a new environment.

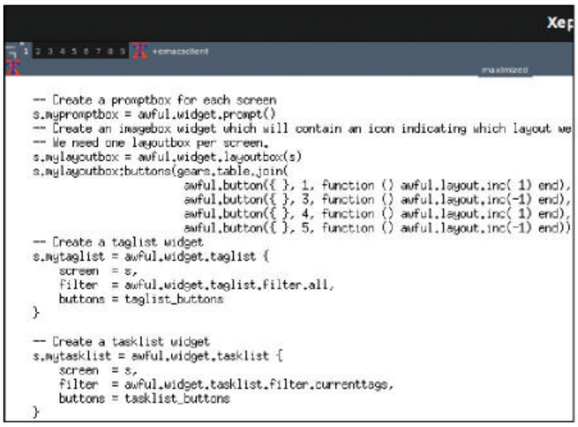
For X11, you need a new server to run on top of the current one. What happens on a regular desktop is that you start one X server, which has **DISPLAY** number 0. To nest, you need a new server that can run on its own and catch another **DISPLAY**. You have a bunch of options for this, *Xephyr* being an excellent choice.

Xephyr creates a new **DISPLAY** variable and acts as if this is a new desktop in most respects. You can use it to start new apps and see how the new environment works. There are a few caveats, one being that your Super key is already busy, so you must use an alternative while testing. Many things will be different once you switch over, so be aware of what your current window manager starts. You may be surprised at how many things Gnome handles on your system.

For Wayland, it is even easier – it nests on its own when you start a window manager.

Window managers

Before you start, you have to read up on a few window managers, choose one and install it. In this example, you are installing the *Awesome* window manager. It is a tiling window manager and has plenty of themes available on GitHub and theming sites. You can choose any that exist for Xorg, because *Xephyr* acts as a new Xorg server. For Ubuntu and Debian, *Awesome* is available as an *Apt* package.



```
-- Create a promptbox for each screen
s.mypromptbox = awful.widget.prompt()
-- Create an imagebox widget which will contain an icon indicating which layout we
-- We need one layoutbox per screen.
s.mylayoutbox = awful.widget.layoutbox(s)
s.mylayoutbox:buttons(gears.table.join(
                      awful.button({ }, 1, function () awful.layout.inc( 1) end),
                      awful.button({ }, 3, function () awful.layout.inc(-1) end),
                      awful.button({ }, 4, function () awful.layout.inc( 1) end),
                      awful.button({ }, 5, function () awful.layout.inc(-1) end)))
-- Create a taglist widget
s.mytaglist = awful.widget.taglist {
    screen  = s,
    filter  = awful.widget.taglist.filter.all,
    buttons = taglist_buttons
}

-- Create a tasklist widget
s.mytasklist = awful.widget.tasklist {
    screen  = s,
    filter  = awful.widget.tasklist.filter.currenttags,
    buttons = tasklist_buttons
}
```

Once you have the basic setup, you can continue tweaking inside the Awesome environment, still assured by your standard setup.

```
$ sudo apt install awesome
```

This software should be in your favourite repository; if it isn't, you can find it at **https://awesomewm.org/download/**. Getting the compiler to work is a little complicated due to the Lua integration method used. For non-developers, it's best to use the packages.

Installing *Sway* is even simpler with your standard package manager. Again, if you want the newest or contribute code, you can find it at **https://github.com/swaywm/sway/releases**.

```
$ sudo apt install sway
```

You may also want to install **swayidle**, **swaylock** and **waybar**. These packages are necessary once you decide to switch, but you have lots of options. You can find a list here: **https://github.com/swaywm/sway/wiki/Useful-add-ons-for-sway**.

You have many other window managers to choose from. Some are designed to look flashy, others to be slimline and fit for a specific purpose. Testing them all requires hours of time and scripting skills, but you can find documentation and community help. Search for BSPWM, Qtile and Cinnamon to start exploring.

QUICK TIP

Plan specific sessions for testing new desktops – it takes time and is exhausting. It is also oddly addictive to try new ways of doing things.

» SWAY AND I3 SHARES

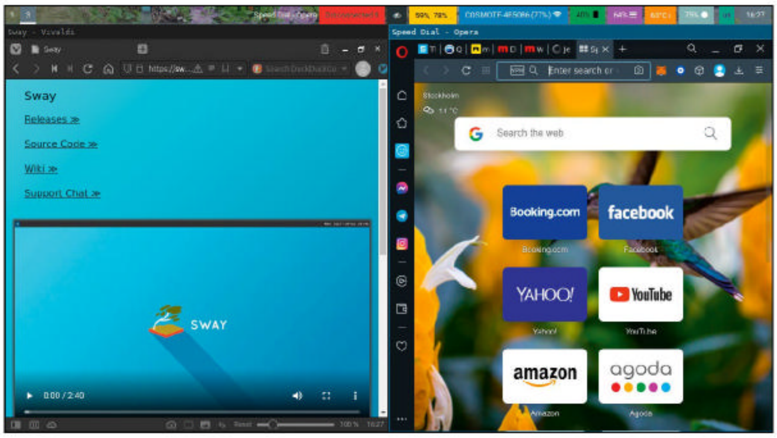
Users of the *i3* window manager should be aware that *Sway* is a clone of the first. It shares the configuration files, which is the main design goal of *Sway*. As you switch over from *i3* under X, you can use the exact same configuration files. You may find some features that are not yet supported in *Sway* but the majority are already there.

Although not recommended, you can even point to your old *i3* files for configuration and run *Sway* out of the box. This makes it easy to make a gradual switch. It also gives you the opportunity to use old shared configurations for new changes to your own setup.

The configuration is well described in the documentation and split into sections, so you can find the info you want. Each component has its own man page. Remember that the status and action bars can be from other systems and many desktop functions are independent of the window manager. Don't discard your choice because some feature is missing, because it may already exist on a general level.

The *Sway* bar (**swaybar**) is the default bar but you can pick others. One version is **somebar**, which has many other features, including clickable areas. With **swaynag**, you can have pop-ups on the desktop, with actions that you can choose from by clicking. This is a powerful way to handle system events and possibly email. It's all up to your imagination and efforts.

»



When adding a few goodies to Sway, the otherwise barren desktop becomes both nice to look at and informative.

QUICK TIP

Your biggest challenge will be keybindings – they tend to be the same on most systems. Being able to lock the keyboard input to the nested desktop is helpful.

When you start running these packages, you will find that they come with less features in order to make them slim. Saving on resources is the first priority, making it necessary to add simple elements on your own. Examples are the action bar and tiling methods.

A window manager is very much a habit – break that habit and you must think carefully every time you want to do something. Fortunately, you'll learn quickly when you switch to a new one. As you try new window managers, though, you end up guessing key combinations and setting everything to your liking.

Tweaking what it looks like can take several attempts, so having your old window manager running while you do this is useful. In this tutorial, you will discover some simple ways to try out the *Awesome* and *Sway* window managers.

To test *Awesome,* we need to nest an X session on top of what you are already running. Using *Xephyr,* you can achieve a likeness and see what happens when you make changes. You can start *Sway* by running `sway`, which starts a nested Wayland session.

» DWL – COMPILE YOUR OWN.

As part of the Suckless (low-overhead design) tools, you can find *dwm*. This is a desktop environment that you compile yourself. Any changes and tweaks are set in the **config.h** file before compiling. No changes are made during execution and you have no configuration files during normal operation.

The idea is to have everything set before you install the software, making the binaries small and efficient. It does make it necessary for users to be at least aware of methods for developing software.

Dwl is a Wayland clone of that system. If you want to try it out, be prepared to handle patches and troubleshoot C code. To be able to build the software, you need to understand how to collect all dependencies for the software. Fortunately, *dwl* only requires two packages: **wlroots** and **wayland-protocols**. Optionally, you may want also **xorg-wayland**; once installed, you need to enable it in **config.mk**.

For an experienced programmer, this is a simple task. Being a green hobby programmer, it is a great way to get started with the whole C/C++ development process. After your first tutorials, this is the way to actually learn the whole process.

Because the designer implemented only a few features, you need to use patches to get everything how you like. Put in the effort, though, and you can move on to your own projects. Once you have passed that hurdle, this is the slimmest desktop you could imagine.

Vital for the use of most window managers, and tiling ones in particular, is the Mod key. You combine this key with another to make things happen. Most common is to have Mod-key+Enter to open a terminal and Mod-key+d to get a menu. As you can probably can guess, this is the Super key in most setups.

When testing it out, you may need to set your ModX-key to Mod1 to use the Alt key, because the Super key will be busy from your main window manager. The Super (or Windows) key is usually Mod4.

The first time you start an application, you may find that it loads in your main window manager, not the nested one. This can happen if you are already running the application.

Xcellent Xephyr

There are several ways to nest an Xorg desktop. You need a new X server to host your desktop and applications. Alas, in Xorg, you need a **DISPLAY** variable set for your screen. Xpra is one, but the designers intended it for remote use, so in this case you use *Xephyr.*

One reason for nesting your sessions is the complexity of settings you have. Another is the wide range of possibilities you have for tweaking your desktop. You need to keep track of your settings and try new ones on a trial and error basis. So, how do you go about this?

The best way is to create all your settings where they will end up when you choose to switch. You can then point to them as you start your new environment.

```
$ cd ~.config/awesome/
$ Xephyr :5 -resizeable -screen 1366x768 & sleep 1 ;
DISPLAY=:5 awesome -c rc.lua
```

As you can see above, you are pointing directly to the configuration file. In the case of *Awesome,* you need to run in the directory where your other configuration files are usually. The **rc.lua** file is yours but if you fetch one from the web, the links are often local under the directory. They do not point to the global position – instead of **~/.config/awesome/themes/** they point to **themes/**.

This makes it simple to switch the entire configuration by changing the directory into another directory. Once you have found your own settings, you have to make small tweaks after you switch to use *Awesome* on your main system.

Even using the nesting system, you won't cover all functions, so make sure you have a backup system for the first few times you run the new one.

Be aware that you must also have the correct version of Lua installed because many configuration files point directly to those binaries.

When the `Xephyr` command starts, you see a new window where the entire desktop shows up. You can lock the keys to the new applications, making it easier to experiment. With the *Awesome* window manager, you can reload the configuration with the key combo Alt+Ctrl+r, making it easy to edit and test repeatedly.

Sway but simpler

Due to the design of Wayland, you can start a new desktop with ease whenever needed. In their work, the designers created the concept of different level compositors. These are: System, which runs from boot;

Session, which runs a user's session; and Embedding, which runs small applications or even a single video from your browser.

You will use *Sway* as a session compositor, making it run nested under your running compositor. Once you decide to stay with *Sway*, you can make it your default from your display manager (GDM) where you login. Other options are available for special systems.

To control what you are doing, look for the configuration file in **~/.config/sway/config**. All of the configuration can be in this file or you can split it into files under **~/.config/sway/config.d**. Run *Sway* from the terminal:

```
$ sway -c ~/.config/sway/config
```

Sway opens like any other application and you can start new applications, which show up on your new desktop. The main reason for doing this, though, is to test the window manager. Play around with the shortcuts and see if the chosen keys suit you. Now, you can also set different configurations and hit $mod-Shift+c to reload the configuration.

This way, you can quickly run through all your ideas while still having a stable main desktop. Here, again, you must be prepared to do more tweaks when you start running it as your main window manager. There will be small details in the configuration that may stop some features.

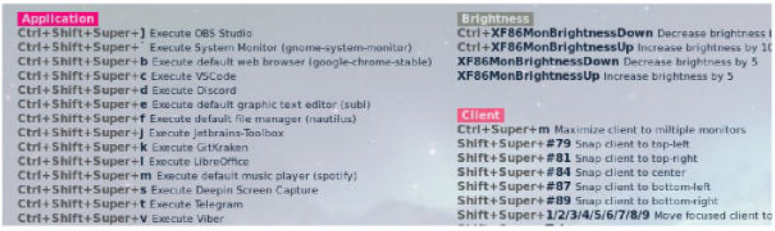
This is all well and good, but there is another case you may want to consider: when you want to run only one application all the time. When would that be? For embedded systems, such as your Raspberry Pi controlling your automation project. A smart home application that does not require Big Tech to know what you're doing is one example. A media player is another, perhaps connecting to your regular TV, giving you a smart home entertainment system.

Any small system you build with a single-board computer can run Linux. You wouldn't choose anything else, would you? All of them have a user interface, some of them even have their own screen. Such a system needs only the application it was meant for and nothing else.

Enter *Cage,* a tiny compositor for Wayland with a few twists. It only opens one window in full-screen mode and exits when that application stops. It also deliberately lacks screen controls and border decorations. All of this creates a system that doesn't have any tools that would capable of doing damage to the system. In this way, the attack surface for hackers is much smaller.

The intention of this software is to make a small system called a kiosk. These systems are the ones where you buy tickets or search for local information or other public services. In such a system, you want only the particular application to run. Often this is written as a web page. In a public space, though, you want only the specific application to run. The safest way to do this is to start it using your initialisation system.

Cage runs any application that you put as the first parameter. In the earlier mentioned scenario, the application is the only one started by your chosen system. Yes, *Systemd* is the industry standard,but there are others!



```
Application
Ctrl+Shift+Super+] Execute OBS Studio
Ctrl+Shift+Super+` Execute System Monitor (gnome-system-monitor)
Ctrl+Shift+Super+b Execute default web browser (google-chrome-stable)
Ctrl+Shift+Super+c Execute VSCode
Ctrl+Shift+Super+d Execute Discord
Ctrl+Shift+Super+e Execute default graphic text editor (subl)
Ctrl+Shift+Super+f Execute default file manager (nautilus)
Ctrl+Shift+Super+j Execute jetbrains-Toolbox
Ctrl+Shift+Super+k Execute GitKraken
Ctrl+Shift+Super+l Execute LibreOffice
Ctrl+Shift+Super+m Execute default music player (spotify)
Ctrl+Shift+Super+s Execute Deepin Screen Capture
Ctrl+Shift+Super+t Execute Telegram
Ctrl+Shift+Super+v Execute Viber

Brightness
Ctrl+XF86MonBrightnessDown Decrease brightness
Ctrl+XF86MonBrightnessUp Increase brightness by 1
XF86MonBrightnessDown Decrease brightness by 5
XF86MonBrightnessUp Increase brightness by 5

Client
Ctrl+Super+m Maximize client to miltiple monitors
Shift+Super+#79 Snap client to top-left
Shift+Super+#81 Snap client to top-right
Shift+Super+#84 Snap client to center
Shift+Super+#87 Snap client to bottom-left
Shift+Super+#89 Snap client to bottom-right
Shift+Super+1/2/3/4/5/6/7/8/9 Move focused client to
```

Keybindings in Awesome, and other window managers, can be hard to remember. To get the reminder page, use "Super+\".

Cagey desktops

To install *Cage*, you can use your package manager. If you have the inclination, the code is available on GitHub: **https://github.com/Hjdskes/cage**. Clone the repo and use *Meson* to build it. Do that if you know how, for the rest of us, install with the `apt` command:

```
$ sudo apt install cage
```

Cage has very few options and they are for window dressing and when you need a dual-monitor config. The only thing you need to do is to choose the application and run it:

```
$ cage sway
```

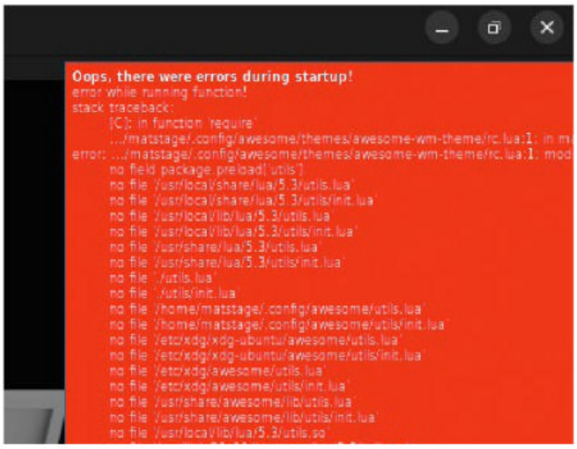
You get the same window, the difference is below the surface; *Cage* is a compositor in itself, so you can run anything in it. You can use it to run apps remotely and have them show up on your desktop. Being a tiny binary, *Cage* is fast and efficient. The point is that you can lock a user in a 'cage', allowing a safer way to allow remote execution via Wayland and its remote back-end.

To run it remotely, it gets more complex. But not much. First, you must set values on the remote system for the wlroots execution – this is described in the documentation. As the designers made *Cage* for the kiosk environment, the most interesting use of this software is single-application boxes.

When switching environment, you need to be aware of both the computer and yourself. The computer must be set up correctly and you must learn to make actions automatically. If you don't, you will waste a lot of time when you have more important things to do. With the right tools, you can run things over the network, too, and have one environment on each computer. LXF

QUICK TIP

The easiest way to hop is between Wayland-based desktop environments. They nest on their own, while X11 requires special software to nest.



```
Oops, there were errors during startup!
error while running function!
stack traceback:
        [C]: in function 'require'
        .../matstage/.config/awesome/themes/awesome-wm-theme/rc.lua:1: in m
error: .../matstage/.config/awesome/themes/awesome-wm-theme/rc.lua:1: mod
        no field package.preload['utils']
        no file '/usr/local/share/lua/5.3/utils.lua'
        no file '/usr/local/share/lua/5.3/utils/init.lua'
        no file '/usr/local/lib/lua/5.3/utils.lua'
        no file '/usr/local/lib/lua/5.3/utils/init.lua'
        no file '/usr/share/lua/5.3/utils.lua'
        no file '/usr/share/lua/5.3/utils/init.lua'
        no file './utils.lua'
        no file './utils/init.lua'
        no file '/home/matstage/.config/awesome/utils.lua'
        no file '/home/matstage/.config/awesome/utils/init.lua'
        no file '/etc/xdg/xdg-ubuntu/awesome/utils.lua'
        no file '/etc/xdg/xdg-ubuntu/awesome/utils/init.lua'
        no file '/etc/xdg/awesome/utils.lua'
        no file '/etc/xdg/awesome/utils/init.lua'
        no file '/usr/share/awesome/lib/utils.lua'
        no file '/usr/share/awesome/lib/utils/init.lua'
        no file '/usr/local/lib/lua/5.3/utils.so'
```

In Awesome, you get a nice list of errors when there are mistakes in the configuration file causing startup problems.

GIMP

Credit: www.gimp.org

Capture and restore old photos and slides

Mike Bedford shows you how to scan, repair and archive old film-based photos, to give your family heirlooms a new lease of life.

OUR EXPERT

Mike Bedford has an extensive collection of film-based photos, both his own and those he inherited, so he's taken quite an interest in preservation and restoration.

QUICK TIP

Be sure to clean as much loose dust, dirt and hairs off your photos before you scan them. Don't use a cloth, though, because that could cause scratches, and don't try using a solvent unless you know exactly what you're doing. Instead, use a compressed air can, but don't hold it too close because that could damaging your photo by freezing it.

Traditional photos, by which we mean the prints, negatives or slides produced by the chemical processes that predated digital photography by well over 100 years, can be remarkably resilient. In fact, it's been suggested that, while we might have inherited photos taken by our grandparents or great grandparents, it's much less likely that today's digital photos will be available to future generations.

Our reference to resilience requires some clarification, because it's by no means guaranteed. For a start, while black-and-white photos are based on a chemical reaction involving silver halides, which are moderately fade-resistant, the dyes used in colour photography were not nearly as forgiving. And second, even in the realm of black-and-white photography, the resilience assumes that the prints or negatives have been well looked after. In reality, this might not be the case. Prints have often been well handled over the decades, so creases, tears, fingerprints and tea stains will have taken their toll. And while negatives have probably not been handled as much, in poor storage conditions – at a non-ideal temperature or high humidity – they could have become dirty, damaged by water or infested by mildew, to name just a few risks.

All this sounds rather depressing, because those old photos might be the only records you have of long-gone family members. However, there is a solution. Here we're going to see how to restore photos that have lost their former glory by using photo-editing software. In fact, you might just be able to make them better than Great Uncle George would have remembered them.

Digitisation

Because your old photos only become eligible for editing once they're in the digital domain, we first need to give some thought to digitisation. If the media is a slide, there probably won't be an associated print, so you'll need to scan the slide, and this involves much the same process as for scanning negatives.

To take an extreme example, the old disk film that was introduced by Kodak in 1982 had a negative measuring just 8x10mm (0.3x0.4 inches), which is very significantly smaller than a typical 4x6-inch print. The relevance of this difference depends on your scanner's resolution. However, if we assume 1,200x1,200dpi, that would result in a difference between 360x480 pixels (less than 0.2MP) for the digitised negative and 4,800x 7,200 pixels (35MP) for the digitised print. With the former being totally inadequate, the recommendation would be to scan the print, even though you wouldn't be achieving the quality that the 35MP figure might suggest, because the optical resolution inherent in the print would probably be lower. Be sure to save your scan in a lossless format such as TIFF and, during the editing process, save all the intermediate results in the same lossless format, so that compression artefacts don't build up. If necessary, though, you can also save a copy of the final results as a JPEG. Also, in the case of a black-and-white photo, save the scan as a monochrome file – greyscale – since it'll give both better results and a smaller file size than if you saved it as a colour file.

If you have a collection of old photos, their condition might leave something to be desired. However, all is not lost.

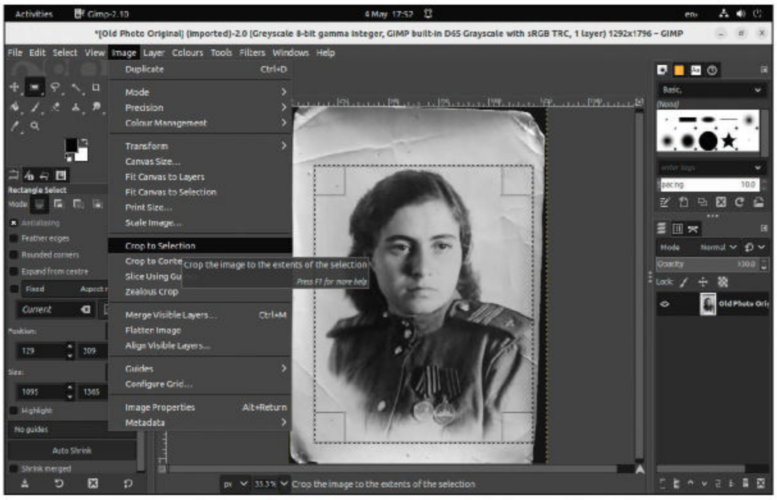
Another scanning option to bear in mind, especially if you decide that your scanner isn't up to scratch, is to use the services of a professional scanning company. Obviously, this will cost a reasonable amount per scan but, depending on how many photos you want scanning, it could be cheaper overall than investing in a new scanner. However, since your prints or negatives will probably be irreplaceable, don't ignore the risk, however small, that's posed by sending them to the scanning company. Using a signed-for postal service offers greater security; better still, use a local company so you can deliver and collect your photos by hand.

Photo editing

Here we're going to be looking at several steps of photo editing, using *GIMP,* but we should point out at the start that you won't always have to carry out all the steps. Each step addresses a particular defect in your old photo, and which of them you need to carry out depends on which of the defects are present. As another word of introduction, you're probably going to be spending quite some time in restoring each of your photos, so be sure to save differently named versions after each step so, if one step goes wrong, you can return to the result of the previous step.

If you have scanned a negative, the first job is to convert it to a positive. Proper transparency scanners can usually convert the image to a positive automatically, but if you use our alternative methods, as described in the boxout *(over the page),* you have to do it manually. In the case of a black-and-white negative, this is easy to do in *GIMP* – just select Colour > Invert. With a colour negative, though, while this is still the first step, the invert function isn't enough, as you'll see when you try it.

You'll notice that the resultant inverted image has a strong cyan colour cast, because physical colour negatives have a red cast. As a next step, therefore, you need to remove the colour cast. So, select the Colour Picker tool and click into the border of the film – the area beyond the image – and this assigns the colour of the border as the foreground colour. Now, select Layer > New Layer and, in the dialog box, select Foreground Colour against Fill With before clicking on OK. Assuming the Layers dialog is displayed, you'll see the new layer above the one with the image. Make sure this new layer is selected and choose Subtract against Mode. The colour of the border is subtracted from all the colours in your image, and this makes it look a lot closer to what you've been expecting. It's probably still not perfect, though, so you need to do some editing, using the usual methods of colour correction – Colour > Auto > White Balance would be a good one to try at first, and might be sufficient. If not, read on for our discussion of restoring colour photos that have become discoloured.



In the case of dog-eared or torn prints, a degree of cropping is your first task.

We're now going to look at the various steps involved in the restoration process – the first few apply to either black-and-white or colour photos, and the final one to colour only. If your photo has suffered physical damage, the worst affected parts are usually the corners and the edges. In that case, you need to crop it to remove any areas that are either totally missing or so badly damaged that they can't be repaired. In the case of damage like tears, the inner parts of which could be more easily repairable than the outer parts, it may be necessary to select a cropped area, which is something of a compromise. In deciding the extent of the cropping, we suggest you crop as little as necessary, without considering the aspect ratio. After all, if you subsequently want to reproduce it with a conventional aspect ratio, it's an easy job to crop it further. To crop, select Tools > Selection > Rectangle Select and then draw a rectangle around the part of

QUICK TIP

Having restored your photo, you might feel that a sepia effect is appropriate, but be sure to keep a non-sepia version, too. This is easily done at Colours > Desaturate > Sepia, although if you're using a black-and-white photo, first convert it to a colour image via Image > Mode > RGB.

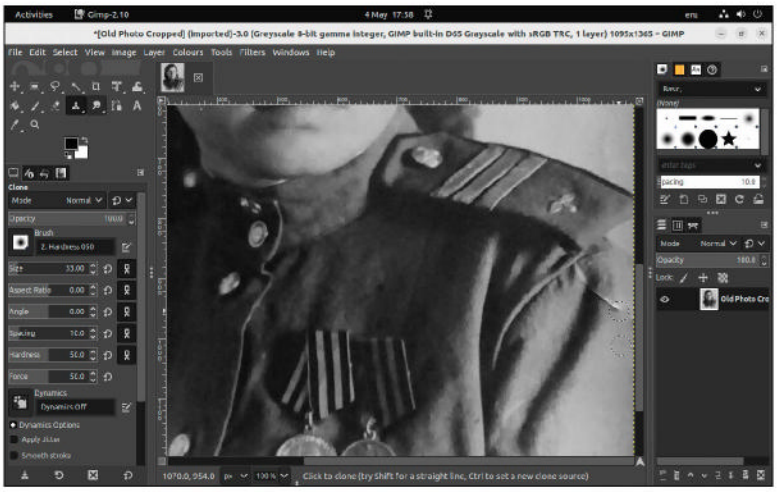
» PROTECT THE SOURCE MATERIAL

CREDIT: Preservation Equipment Ltd.

Having digitised and subsequently restored your family heirlooms, you probably won't want to abandon the source material. After all, a 100-year-old family album has a unique appeal and sentimental value, even if the photos it contains aren't as good as your edited digital equivalents. What's more, you might feel that slides, negatives and prints have historical interest, and that it would be a shame to totally lose an appreciation of old-school photography. Unfortunately, however, those old photos will continue to degrade, and it's inevitable that, sooner or later, they will be of no value whatsoever. There is some good news, though, namely that with proper care, you can slow down their deterioration.

That care all comes down to storage, so use appropriate containers. Ordinary paper contains acid, so store paper or photographic prints in envelopes, folders and/or boxes that are acid-free and intended for storing valuable and old documents. Make sure the folders or envelopes are a suitable size for their contents, as documents that are free to move excessively may get damaged. Ideally, make sure prints and negatives are separated using sheets of acid-free paper, so they don't stick together. And store the boxes or envelopes vertically, using acid-free padding to prevent the documents from slumping if any boxes aren't full. Suitable document archiving boxes are available from **www.preservationequipment.com**. Using proper containers will keep your photos dark, which is good, but also store them in a medium-low humidity, moderately low temperature environment.

»



*[Old Photo Cropped] (imported)-3.0 [Greyscale 8-bit gamma integer, GIMP built-in D65 Grayscale with sRGB TRC, 1 layer] 1095x1365 – GIMP

1070.0, 954.0 px 100 % Click to clone (try Shift for a straight line, Ctrl to set a new clone source)

You'll probably be using the clone tool extensively to hide scratches, creases and miscellaneous blemishes.

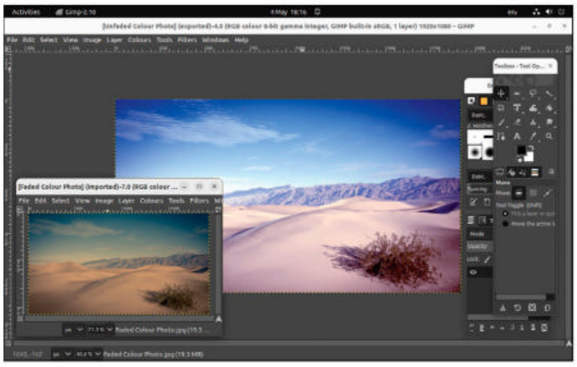
the image you want to retain. Now, all you have to do is select Image > Crop To Selection.

Next up we're going to see how to remove, or at least reduce the visibility of, imperfections such as creases, scratches and miscellaneous spots or blotches. We suggest you use the Clone tool, which enables you to overwrite an area of the image with whatever colours and textures are in a nearby area of the same size. Select that tool from Tools > Paint Tools > Clone, hold down the Ctrl key and click on the area you want to clone from, then move to the area you want to clone to, and either click once or click and move to clone over a larger area. We recommend single clicking, at least to start. It's probably best to use a degree of transparency, and perhaps use a brush that doesn't have a hard edge. You can select these and various other tool attributes, including the size of the clone area, in the Tools Options dialog, which you have to enable if it's not already visible. In some instances, you might prefer to use the Heal tool, also under Tools > Paint Tools, which has a similar function, but we'll leave you to read up on the differences.

While clone and heal are the best ways of removing defects, if there so many it would take you for ever, you might like to experiment with noise reduction, which you can find at Filters > Enhance > Noise Reduction. While this can remove a lot of the hard work, especially for small specks, it also makes the whole image appear less sharp, so use it sparingly – don't choose too high a value for Strength in the dialog.

So far we've seen how to deal with specific defects that you'll encounter in old photos, but we also need to consider that the image as a whole might have lost some of its vibrancy, due to fading or decolourisation. In the case of a black-and-white image, you can probably make a significant improvement by increasing the contrast, although in the case of a colour photo, we suggest you skip to the next paragraph. The simplest way of increasing the contrast is found at Colours > Brightness-Contrast – and you'd probably need to adjust the brightness, too – but you have more control in the Adjust Colour Curves dialog, which you can access from Colours > Curves. The dialog shows how the original range of brightness values are mapped on to the edited values, and the initial straight diagonal line defines no change. To use this tool, you edit the shape of the line by clicking and dragging, and a good first choice is to change the straight line to an s-shaped curve, by which we mean the steepness is decreased at the lower-left and upper-right ends. Trying it out yourself will quickly show you how it works.

And finally, if you're working with an old colour image that was scanned from a print, you'll probably find that the colours look faint and unnatural. This is because the dyes in the print will have faded over time and, to make matters worse, some of the primary coloured dyes will have faded more than others. *GIMP* has an automated tool that can make a huge difference and, depending on the degree of fading, might be sufficient in itself. Select Colours > Levels and then, in the Adjust Colour Levels dialog, ensure that Preview is checked and click on Auto Input Levels. Hopefully you'll see a significant improvement but, if you want to try to make it even better, click on Edit These Settings Using Curves. This causes the Adjust Colour Curves dialog to appear, which we investigated in the previous paragraph, but now, in addition to the Values curve,



Activities
Gimp-2.10
4 May 18:16
[Unfaded Colour Photo] (exported)-4.0 (RGB colour 8-bit gamma integer, GIMP built-in sRGB, 1 layer) 1920x1080 – GIMP
File Edit Select View Image Layer Colours Tools Filters Windows Help
Toolbox - Tool Op...
[Faded Colour Photo] (imported)-7.0 (RGB colour ...
File Edit Select View Image Layer Colours Tools Filters Windows
Faded Colour Photo.jpg (19.3 MB)

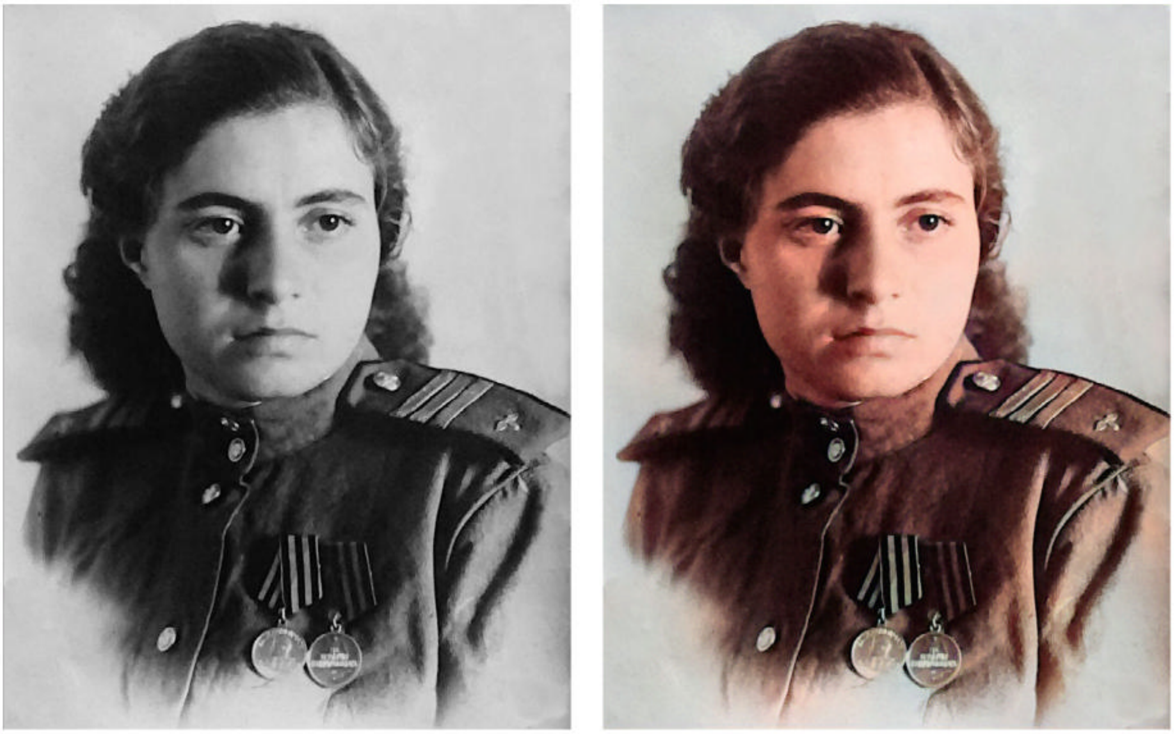
Even by just using GIMP's automatic colour correction tool, you can make a huge difference to badly faded colour photos.

» OTHER FILM SCANNING OPTIONS

If you have a lot of film-based media to scan, a proper solution, such as a scanner capable of scanning transparent material, is surely the way to go. However, if you only have a few negatives or slides, there is an alternative solution which, while certainly not as good as a scanner, might just provide you with acceptable image quality.

That solution is to photograph the slide or negative, ideally with a decent camera or, failing that, using a good quality phone. You need to backlight the film with a diffuse white light source – our solution is to use a tablet, placed horizontally, displaying just a plain white background. Arrange for the slide to be held several centimetres from the screen, otherwise you'll capture the pixel matrix of the screen superimposed on your film-based image. Hold the negative or slide in a cardboard window to mask out the surrounding area of the screen, so the light metering isn't adversely affected. Ideally, mount your camera or phone on a tripod. If you don't have a tablet, you could use your PC's monitor, but fixing the slide or negative in its cardboard window at a suitable distance from the screen would be trickier.

CREDIT: www.publicdomainpictures.net/en/view-image.php?image=109181



There's no guarantee that the colours will be correct but, in our experience, the tool at Img2go makes a pretty impressive job of colourising restored black-and-white photos.

CREDIT: Original photo: Adam Jones, CC BY 2.0, www.flickr.com/photos/adam_jones/26683526870

you'll see curves for Red, Green and Blue. Fine-tune any or all of these curves by selecting the appropriate one against Channel.

Before moving on from the subject of editing your scanned image with the aim of making it as good as new, let's take a quick look at how you could make a black-and-white image better than new. Not everyone will agree with that perception, and we certainly wouldn't want to dismiss the appeal of black-and-white photography, but you might like to experiment with turning your old black-and-white photo into a colour one. You could certainly do that manually in *GIMP*, and while that would involve quite a lot of work, it'll probably give the best results, so long as you know what colour everything should be. However, you could try an automated colourisation tool, and while recognising that it won't have any knowledge about the colour of your great grandmother's dress, the results can be pretty impressive. The tool we tried is hosted online, and you can find it at **www.img2go.com/colorize-image**. Just upload your restored black-and-white photo, click on Start, and prepare to be amazed. We also noticed that, if you upload a badly faded colour photo, it makes a good job of restoring the original colours – indeed, if our faded photo was typical, it performs better than *GIMP's* Auto Input Levels in its Adjust Colour Levels dialog.

Archive your results

It's quite possible that your photo-editing skills have turned back the clock, and restored your old photos to the way they looked decades ago. But don't forget that the clock will start ticking again straight away, and unless you look after your edited photos, they'll start degrading. So, if you want to pass them on to future generations, you'll need to look after them. And here we're not talking about inkjet prints of your restored photos, because that wouldn't be a good way of preserving them for any length of time, but about the edited digital images.

The way in which your digital images degrade is entirely different from the way the initial photographs degraded, and potentially more catastrophic. Prints and negatives degrade gradually, but their digital equivalents are more likely to degrade instantaneously. That loss could be the result of a disk crash, and this emphasises the importance of archiving your edited photos. As with any data, there are two main options. First, you could archive into the cloud, but remember that it's not uncommon for companies to go out of business, so there's no guarantee that your online archive will be available to future generations. Second, you could archive to external local media, but again there are issues to consider. First, digital data tends to be temporary in nature – in other words, having written your photos to a DVD-R disk, for example, there'll be a limit to how long into the future you'll be able to read it. So, read up on the longevity of your media and rewrite your archive to new media well before the safe period expires. But there's more, as you'll be aware if you happen to have any old floppy disks lurking in a drawer. Quite apart from whether the data is still intact, you probably don't have a floppy disk drive on your PC, and the same will eventually be true of today's generation of optical disks. Sure, you'll be able to find service providers who'll be able to read them for some time into the future, but that probably won't be a long-term solution. So, be sure to rewrite your archive to new types of media as and when necessary. Since neither archiving approach is without its problems, it would be wise to adopt both a cloud-based and a local external media strategy to provide extra peace of mind. LXF

QUICK TIP

You might like to print some of your restored photos on glossy paper and mount them in an old-fashioned album. Despite the recent trend towards printed photo books, you can still buy albums, often of a traditional design, even down to the use of corners to hold the photos in place.

ARDOUR

Credit: https://ardour.org

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

Compose using the ultimate tracker

Michael Reed explores some common music-making workflows in Ardour 7, the premier Linux digital audio workstation application.

OUR EXPERT

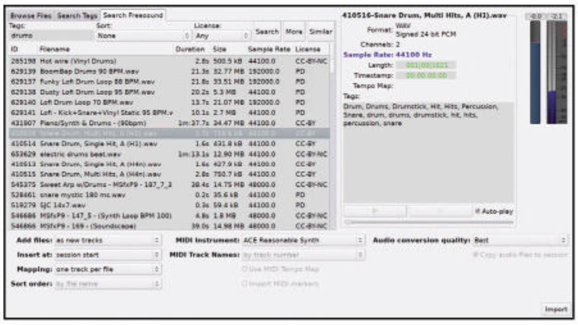
Michael Reed once simulated a reverb chamber by putting a guitar amplifier into an empty wardrobe with a microphone. Be aware of this if you are considering inviting him to your house.

Ardour is an open source application for recording and composing music. It can work with audio tracks such as recordings of instruments and it can also handle MIDI tracks, which are notes on a grid that trigger plugins. Speaking of plugins, it can handle all the major plugin formats both for instruments and effects. In the final stage of the process, it can output a finished audio file in a format such as WAV, MP3 or FLAC. It's mainly seen as a music program, but it can also handle other types of audio projects, such as podcast recording and editing.

Ardour is capable of a lot, but it's a complex piece of software. We're going to take you through a few common workflows. And as we do, we'll shine a light on some of the features that were introduced in *Ardour 7*.

Launching Ardour

Every time you launch *Ardour*, you are presented with the Session Startup dialog. From here, you can select a session to load, choose from a list of recent sessions or create a new session. Click on New Session to create a blank session. The next dialog gives the project a temporary name based on the date.



Browse Files | Search Tags | Search Freesound

Tags: drums — Sort: None — License: Any — Search — More — Similar

ID	Filename	Duration	Size	Sample Rate	License
285198	Hot wire (Vinyl Drums)	2.8s	500.5 kB	44100.0	CC-BY-NC
629139	BoomBap Drums 90 BPM.wav	21.3s	32.77 MB	192000.0	PD
629137	Funky Lofi Drum Loop 88 BPM.wav	21.8s	33.51 MB	192000.0	PD
629138	Dusty Lofi Drum Loop 95 BPM.wav	20.2s	5.3 MB	44100.0	PD
629140	Lofi Drum Loop 70 BPM.wav	13.7s	21.07 MB	192000.0	PD
629141	Lofi - Kick+Snare+Vinyl Static 95 BPM.v	10.1s	2.7 MB	44100.0	PD
431907	Piano/Synth & Drums - (90bpm)	1m:37.7s	34.47 MB	44100.0	CC-BY
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
410514	Snare Drum, Single Hit, A (H1).wav	1.6s	431.8 kB	44100.0	CC-BY
653629	electric drums beat.wav	1m:13.1s	12.90 MB	44100.0	CC-BY-NC
410513	Snare Drum, Single Hit, A (H4n).wav	1.6s	427.9 kB	44100.0	CC-BY
410515	Snare Drum, Multi Hits, A (H4n).wav	2.8s	750.7 kB	44100.0	CC-BY
545375	Sweet Arp w/Drums - MSfxP9 - 187_7_3	38.4s	14.75 MB	48000.0	CC-BY-NC
528461	snare mystic 180 mc.wav	0.2s	35.6 kB	44100.0	PD
519279	SJC 14x7.wav	0.3s	59.4 kB	44100.0	PD
546686	MSfxP9 - 147_5 - (Synth Loop BPM 100)	4.8s	1.8 MB	48000.0	CC-BY-NC
546866	MSfxP9 - 169 - (Soundscape)	39.0s	14.98 MB	48000.0	CC-BY-NC

410516-Snare Drum, Multi Hits, A (H1).wav

Format: WAV Signed 24 bit PCM
Channels: 2
Sample Rate: 44100 Hz
Length: 001|00|1021
Timestamp: 00:00:00:00
Tempo Map:
Tags: Drum, Drums, Drumstick, Hit, Hits, Percussion, Snare, drum, drums, drumstick, hit, hits, percussion, snare

Add files: as new tracks
Insert at: session start
Mapping: one track per file
Sort order: by file name
MIDI Instrument: ACE Reasonable Synth
MIDI Track Names: by track number
Audio conversion quality: Best

Browsing the Freesound library thanks to Ardour 7's integration of it.

The next dialog is concerned with the audio system you want to use, and some experimentation might be needed here. ALSA connects you directly to the soundcard, but it means that you can't use other sound applications at the same time (this includes the web browser). JACK is usually the better option, but it

QUICK TIP

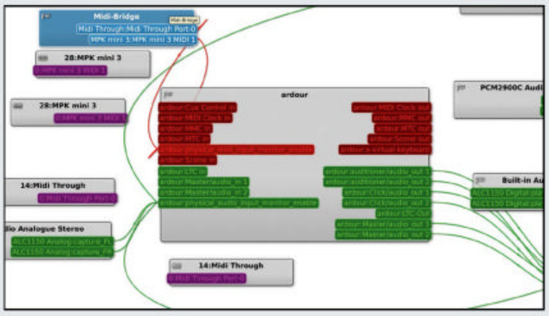
If you notice that the audio output of Ardour is delayed, adjust the buffer size (Window > Audio/MIDI Setup) – 128 works well on most systems, but experiment.

» CONFIGURING ARDOUR

As a collection, the Linux sound systems have a reputation for being complicated, but things might be set to calm down in that area thanks to the introduction of PipeWire, a system that aims to replace some of the others. It offers emulated interfaces for systems such as JACK and ALSA, and you can route audio around between programs and hardware inputs and outputs.

Ardour is open source software, but there is a catch as the developers expect you to make either a regular or a one-off payment if you want to download a premade binary. To be fair, to get the program and all the point updates until the next major release, you can pay what you like.

To get *Ardour* for free, you can build it from the source code, and that's our preferred option as distribution builds tend to be out of date. It uses the WAF build system and the website provides build instructions (**https://ardour.org/development.html**). We regard it as a medium difficulty build, and you're probably going to have to spend some time hunting around your distro's package repository for the needed dependencies.



Using Pipewire tool Qpwgraph to determine what interfaces are present and what is connected to what.

sometimes requires some extra setup. Whichever option you choose, click Start to begin using *Ardour.*

Playing with loops

Ardour 7 introduced the option of working with the Freesound sound library, and working with loops downloaded from this source is as good a way as any to familiarise yourself with how *Ardour* works. This holds true even if you don't, eventually, want to work with loops.

An account is needed to use the Freesound service, but it's free to create one. Access the site (**https://freesound.org**) and follow the prompts from the Login link. The first time you access Freesound resources from within *Ardour* (from Session > Import), the program contacts the Freesound website, which produces a code number that you can cut and paste back into *Ardour.*

Note that the Import dialog window is the one you would use to import audio files from your local hard drive. However, in this case, we'll work with material from Freesound.

Within the Import dialog, select the Freesound tab and search for the word 'drums'. Scroll through the list and look for something that seems suitable. You're looking for something that has a duration of around two to 10 seconds, as we need something that can be looped. You can preview the sound by pressing the play icon within this dialog. When you've found something you like the sound of, click on Import. This creates a track within *Ardour* and imports the downloaded material. Switch on the metronome (top icon panel) and press play. The loop plays while the position cursor moves over it.

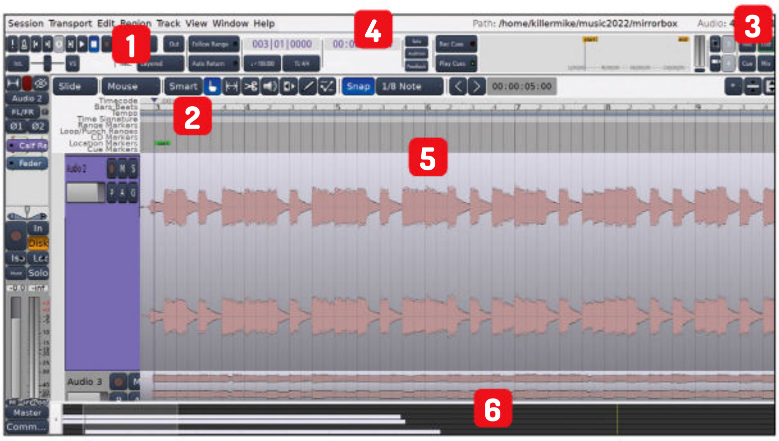
A bit of a stretch

Ardour defaults to a tempo of 120bpm (beats per minute), and it's a good thing if the drum loop you've added doesn't match that tempo as it's an opportunity to use *Ardour's* audio stretching facilities. For example, we chose a drum loop with the filename **drums_98_swing_full**, and we were able to deduce that it was a drum beat with a tempo of 98bpm. Fortunately, *Ardour* has facilities to speed up and slow down a drum loop without altering the pitch.

Select the Stretch Tool in the toolbar. Hover the mouse anywhere along the drum loop container on the timeline and then drag the edge so that it matches one of the marker lines in the arrange window. A dialog pops up telling you how much the duration is being altered. For instance, if you dragged the edge of the container so that it fills one bar, our example loop would be sped up so that it fills 81.6% of the space. There are also options in this dialog to control the algorithm that is used for the pitch shifting, and we recommend choosing Unpitched Percussion With Stable Notes for material such as a drum loop. If you click on OK, the conversion takes place and the drum loop should now cover exactly one bar and conform to a 120bpm metronome click.

The other way we could have taken things would be to increase the length of the drum loop so that it fills exactly two bars. If we do this, the dialog tells us that this means increasing the length of the container to 163.3% of its original size. The problem here is that stretching it by such an extent introduces massive distortions to the sound. Try it, but you'll probably find that such a sound isn't really useful. *Ardour* can only stretch things by about 20% without greatly damaging the sound quality.

EXPLORE THE ARDOUR INTERFACE



Session Transport Edit Region Track View Window Help
Path: /home/killermike/music2022/mirrorbox
Slide
Mouse
Smart
Snap
1/8 Note
00:00:05:00
Audio 2
Audio 3
Master
1
2
3
4
5
6

1 The transport bar Play, record, return to start, metronome toggle and more.

2 Editing mode toolbar Hover to find out what they all do. Grab Mode (Select) is usually the best starting point.

3 Window selection You can select a window here – Edit, Rec, Mix and Cue.

4 Main toolbar The top toolbar is common to all of the Ardour windows.

5 Edit window Similar to other DAWs, the Edit window is where you arrange audio and MIDI regions.

6 Tune layout The combined scroll and zoom tool previews the layout of your tune.

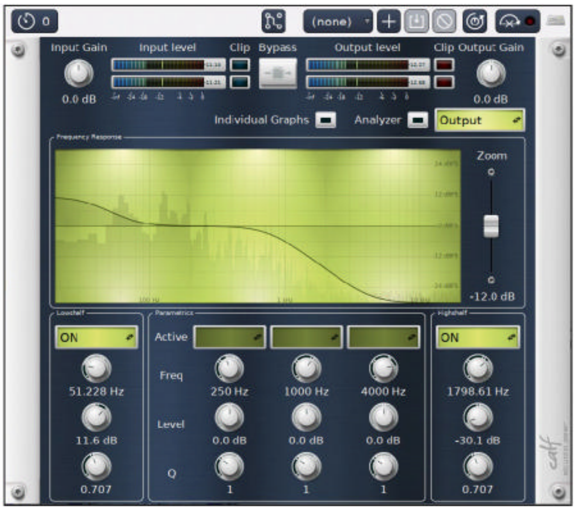
Loop the loop

Generally, you want a drum loop to play over and over, and you do this by making multiple copies of the loop that are spread around the song. To do this, click on the drum loop within the arrange window (use Grab Mode in the toolbar). Press Alt+D to open the Multi-Duplicate dialog. Change the number to 15 and press OK. This means we have a total of 16 copies of this drum loop.

Another way you could have done this is to click once on the drum loop and keep pressing Ctrl+D (rather than Alt+D) to create more duplicates until you have enough. Use Ctrl+click to multiple select the loops, then combine them (Selected Regions > Edit > Combine). Combined in this way, they can be moved

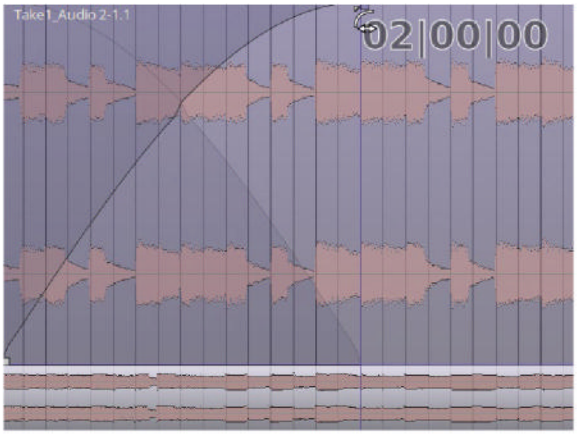
QUICK TIP

The requirement of a donation in return for the prebuilt package won't suit all free software advocates. But a well-funded project benefits everyone, and you can get Ardour for free by building from source.



The Calf effects are excellent, and they cover most of the basic studio effects that you're likely to need.

»



Editing a fade-in is one of the simplest edits to make. Drag the small square in the top-left corner of a region to adjust.

and duplicated as one solid group rather than individual copies that would have to be multiple selected.

Add effects

It's not uncommon to add an effect such as reverb to a drum loop, and there are a few ways you can add an effect to a track using the *Ardour* interface. You can do this while still in the Edit window, but it's a good opportunity to become accustomed with the Mix window, *Ardour's* channel mixer. Select this by clicking the Mix button in the top-right hand of the screen.

For the moment, you have only one track in *Ardour,* which is probably named after your imported drum loop. To add an effect to that track, right-click on the top half of the Fader button that is part of that mixer strip and select New Plugin > Plugin Selector. This opens *Ardour's* plugin manager.

Use the search box to search for 'reverb'. If you've taken our advice and added the Calf Plugins, you can add the reverb from that set. Double-click on Calf Reverb from the list. It's a useful plugin, so you might like to tick the Fav box next to it for easy access next time. Double-clicking adds it to a list of plugins in the bottom right-hand side of the dialog that are going to be added to the track. Click on the Insert Plugin(s) button to actually add the plugin to the track.

If you now play back the song (you can do so without leaving the Mix window), you should hear a reverb effect on the drum loop. You can alter the parameters of the reverb effect by double-clicking on it on the mixer strip. The reason that we right-clicked on the top half of the Fader button in the mixer strip is that we wanted the reverb effect to be placed before the fader. If you've accidentally added it after the fader, simply drag it to a position before the fader. You can reorder any of the effects in this way. If the reverb was after the fader, the fader would alter the amount of signal that is going into the reverb effect. Generally, you want the fader to control the volume of everything on a track including an effect such as reverb.

QUICK TIP

Many of the functions in the Ardour main menu can also be found in the right-click pop-up menu. The pop-up menu only shows operations appropriate to the region being clicked on.

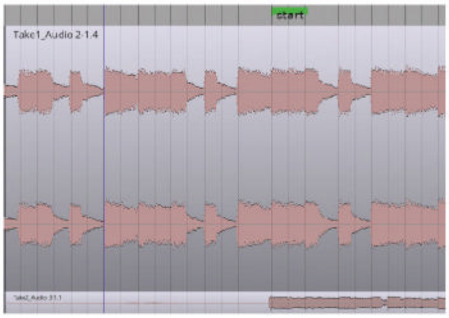
Recording a guitar

Ardour can handle MIDI sequencing, but its specialism is working with audio, and it can record audio tracks from a source outside of the computer. The procedure is much the same if you are working with an audio source such as a microphone or an instrument such as a guitar plugged into an audio interface. How you connect such a source to your computer depends on the type of audio interface you are using.

Go back to the Edit page, if you're not already there. Create a new audio track by right-clicking on the track panel area. In the menu that pops up, choose Audio as the track type. If you're coming straight into your audio interface, you might prefer to create a mono track.

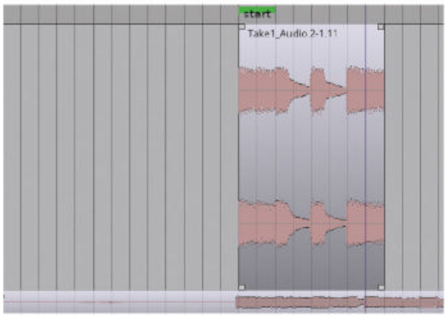
Before, we manipulated the track settings from the Mix window, but you'll note that, when working in the Edit window, the currently selected track has its mixer strip to the right of it. Hover over the mixer strip's buttons to discover which is the input selector, and select the audio input that you want to use. If you're using a microphone plugged into your audio adaptor or an instrument such as a guitar, make some noise to see if the track level meter begins to jump around. When

EDITING AUDIO



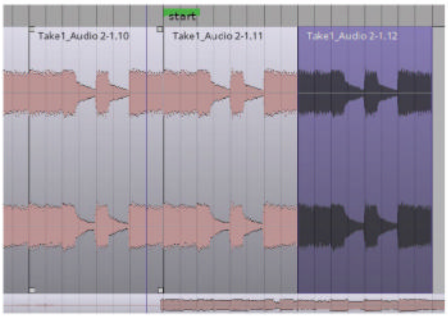
1 Select the audio

You've played a guitar riff four times and you've decided that the third repetition is perfect. Now you have to repeat that riff over and over. Put yourself into Slide Mode rather than Ripple Mode in the toolbar. Make sure you're in Grab Mode (toolbar or press G) in order to select the correct region and click anywhere along the region that contains the guitar recording.



2 Start cutting

In this case, the guitar part is four measures (bars) long. So, we need to cut out the area that's between measures three and four. Select Cut Mode (toolbar or press C). Cut twice, once at the third measure and then once at the fourth. Go back to Grab Mode and delete the 1-3 section (click on it and press Delete) and the 4-5 section.



3 Move and duplicate

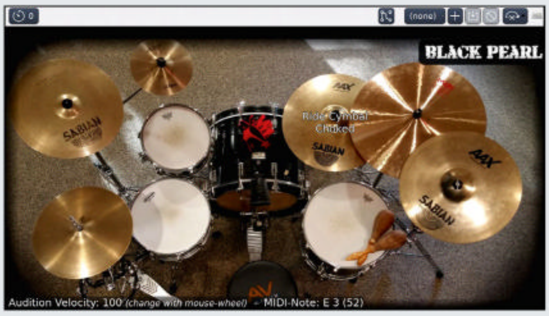
While still in Grab Mode, select the good take of the guitar riff, which should be sitting at measure 12. Move it by dragging it to section one, the start of the song. With it still selected, duplicate it three times (Ctrl+D three times). Join all four sections into a new region by multiple selecting them (Ctrl+click) and then joining them (Selected Regions > Edit > Combine).

RECOMMENDED PLUGINS

We highly recommend the Calf Studio Gear (**https://calf-studio-gear.org**) collection of audio effects. These come in the Linux native LV2 plugin format and are available in the repository of most Linux distros. These effects cover all the basics, such as reverb, chorus and compression, and the quality is superb. The user interfaces of each effect are also excellent and typically feature colourful real-time displays complete with all sorts of useful meters.

When it comes to synthesiser instruments for the MIDI side of *Ardour,* SurgeXT (**https://surge-synthesizer.github.io**) is one of the best options. It was formerly a commercial product that has been made open source, and it's under constant development from the community. We think the LV2 version of the plugin works better in *Ardour* than the VST one.

If you like realistic drum sounds and you prefer MIDI composing to using drum loops, consider the x42 AVL drum plugin (**https://x42-plugins.com/x42/x42-avldrums**). This was made from samples of a real drum kit with hits at various velocities, and there is even a selection of different drums. Specifically, it exposes a lot of routing possibilities in *Ardour*.


BLACK PEARL
Ride Cymbal Choked
Audition Velocity: 100 (change with mouse-wheel) - MIDI-Note: E 3 (52)

The x42 AVL drum plugin has a real-time display of the drum hits and allows for volume adjustments.

you are sure that your instrument or voice is routed to the track, you can begin to make a recording.

To record in *Ardour,* you must arm the current track. You do this by clicking on the red, circular record icon on the track. Secondly, you must place *Ardour* itself into record mode by clicking on the Record icon in the transport bar. Once *Ardour* is in record mode, it behaves as normal other than the fact that it records audio to any armed tracks. This means that you can record to more than one track at once if your audio interface has the inputs.

When you are ready to begin, press play as normal and the recording starts. If you make a mistake, stop the recording (on the transport bar or press Space).

MIDI editing

We'll close this overview with some remarks about MIDI editing. *Ardour* is better known as a multitrack audio recorder and editor, but it has basic MIDI sequencing features as well. One of its strengths in this area is that its MIDI and audio editing facilities are quite similar, so most of what you've learned about audio recording and editing can be transferred across.

As when creating an audio track, you can create a MIDI track by right-clicking in the track panel area. This time, select MIDI rather than Audio as the track type. To hear any sound from a MIDI track, there must be an instrument plugin on the track. You can select such a plugin from the list of detected plugins when creating a track. If you create a MIDI track without a plugin, you can add one later using the Plugin Manager in the same way we did with the reverb plugin. As with audio, you can add effect plugins after the instrument plugin.

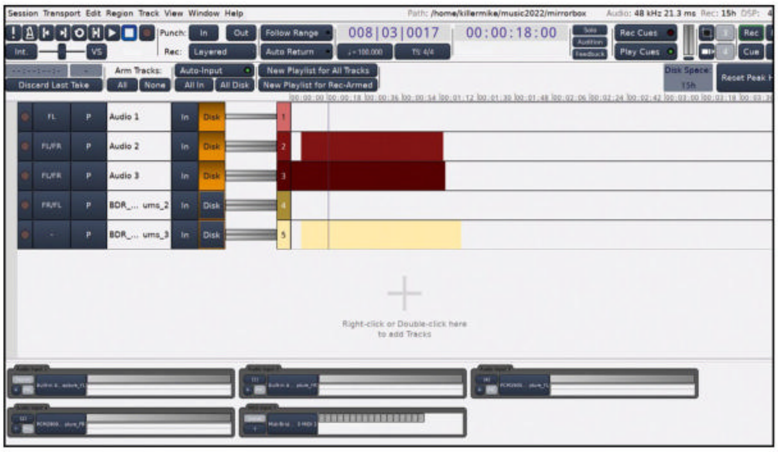
The MIDI editing in *Ardour* is done inline just like the audio editing. Double-click on the lower part of the track in the track panel area to expand the size of the track line so you can see the notes. You can add notes manually using the mouse by selecting Draw Mode from the toolbar.

Recording works in the same way as for audio. Arm a MIDI track and put *Ardour* in recording mode. When you press play, any MIDI input is added to the region.

Exporting projects

When you've finished your song by recording it, editing it and mixing it by adjusting the track levels and adding effects to them, it's time to export it as an audio file. The simplest way of doing this is to use the quick export (Session > Export > Quick Audio Export).

Hopefully, we've taken you through a common *Ardour* workflow to create a song from scratch. One advantage that *Ardour* has over some other music programs is that the many elements of the user interface are shared between working with MIDI and audio. So, every time you learn a new trick in one area, it can be used when working with the other. Best of luck exploring *Ardour's* extensive features! LXF



The Recording window is a stripped-down overview of your tracks for hassle-free recording.

TAKE YOUR LINUX KNOWLEDGE FURTHER

Enhance your knowledge with in-depth projects and guides

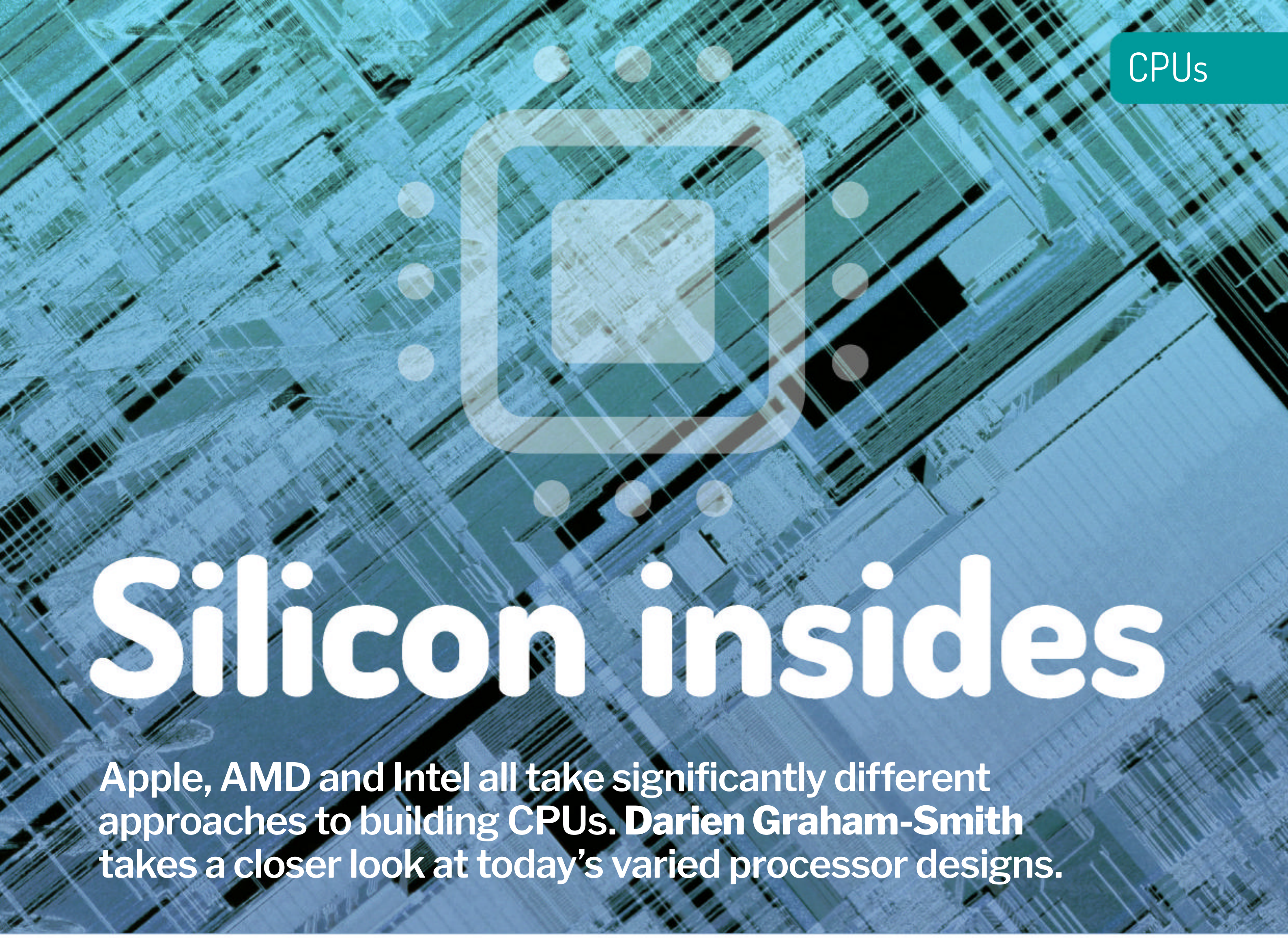
HACKER'S MANUAL 2023
FULL OF EXPERT TIPS & ADVICE
ADVANCE YOUR LINUX SKILLS
• THE KERNEL • NETWORKS
• SERVERS • HARDWARE • SECURITY
148 PAGES OF TUTORIALS
ENHANCE YOUR KNOWLEDGE WITH IN-DEPTH PROJECTS AND GUIDES
ON SALE NOW
Hacker's Manual 2023 Fourteenth Edition
ISBN 978-1-80308-729-0
FUTURE
FOURTEENTH EDITION

FUTURE

Ordering is easy. Go online at:

magazinesdirect.com

Or get it from selected supermarkets & newsagents



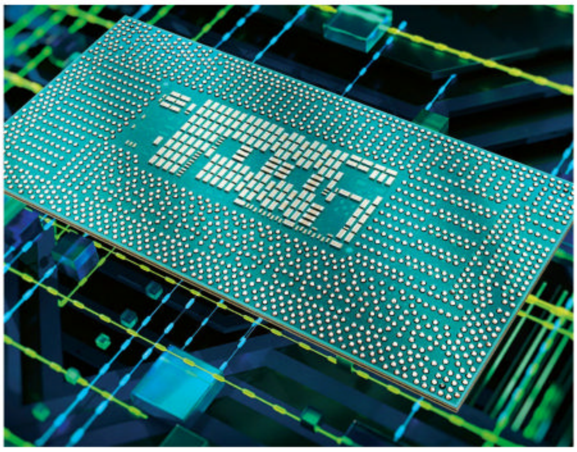
Silicon insides

Apple, AMD and Intel all take significantly different approaches to building CPUs. **Darien Graham-Smith** takes a closer look at today's varied processor designs.

No one would have believed in the last years of the 20th century that the processor world was being watched keenly and closely by intelligences greater than Intel's. Yet across the gulf of the Atlantic in a country called the UK, intellects vast and cool regarded the processor market with envious eyes, slowly and surely drawing their plans against Intel…

History is a funny old thing, when IBM was part-picking to build the first PC, it could have gone with its own IBM 801 RISC processor, but the budget insisted on the Intel 8088 and history was set: every PC would be running an x86-compatible processor.

Technically, anyone could design and manufacture an x86-compatible processor, but legally (we're looking at you China), Intel owns the patents to the instruction set and has to license it for that to happen. If a company has ever produced a design or manufactured an x86 processor, it's because Intel (or a court) allowed it to. AMD has a complete cross-patent licensing agreement with Intel, so the two companies don't end up suing each other into oblivion.



Intel's 12th-gen Alder Lake CPUs combine E-cores with P-cores.

Image credit: Intel

The point is there's not much competition for Intel in the market. Even today, with AMD commanding a record share, that's just 30% of the consumer market. AMD itself said it was aiming for just 10% of the server market in 2020 and while it's doing well its EPYC server share has levelled off at around 18%.

It's fine to lament the lack of competition, but what can possibly change to break the status quo? The big recent announcement is that Apple will start to move away from Intel-based processors and switch all of its hardware to its own design of Mx processors. Apple's not talking about just laptops or low-end iMacs, but even its high-end workstation offerings that use the Intel Xeon. It's a bold statement, so how did it manage this? Let's explore the differences between their approaches, and what that means in practice.

The x86 and ARM processor platforms do the same basic job, but they do it in different ways. Their internal logic is wired up in different arrangements, with different configurations of internal data registers and different sets of hard-coded instructions. At a fundamental level, they run programs in different ways and use different code.

On the x86 platform, the internal structure and instruction set of the processor is ultimately based on that of the Intel 8008, an 8-bit CPU that debuted in

»

From left to right: Apple's M-series CPUs have been using a 5nm process since 2020.

All of Intel's x86 chips use the same underlying architecture.

AMD's chips use the same core x86 architecture.

1972. In fact, machine code programs written for that chip can still be assembled and run on the latest processor from Intel or AMD.

Naturally, though, the hardware has evolved considerably since then. After the 8008 came the 8088, and then the 16-bit 8086, which powered the original IBM PC. In the '80s, this was followed by the 80186, 80286 and so forth – hence the "x86" moniker.

Through the generations, new features have been introduced to support multitasking and virtual memory; support has also been added for 32-bit and 64-bit operations, enabling computers to work efficiently with huge data sets and massive amounts of RAM. A series of extensions accelerated specific tasks such as graphics processing, virtualisation and data encryption.

Apple's processors are based on the competing ARM architecture. This originated at Acorn Computers in the mid-'80s, at a time when the company was looking to create a successor to the hugely popular BBC Micro. Rather than buying chips from an external supplier, as it had with its previous home computers, the company set out to design a new processor that would outperform existing rivals. And it succeeded – at its launch, the ARM-based Acorn Archimedes was the most powerful home computer money could buy.

> **LOW-RISC STRATEGY**
> "The RISC philosophy aims to make a CPU as simple as possible, by reducing it to a bare minimum of basic functions."

Today, the ARM platform is owned and developed by Arm Limited in Cambridge, and like x86, it's continued to grow since its inception. Successive versions of the platform have added 64-bit support and numerous extensions to speed up common mathematical operations – including, in the latest ARMv9 release, security and AI features.

RISC vs CISC: The eternal rivalry

While ARM processors can do anything x86 can, they have different strengths and weaknesses, as they follow a different design philosophy, known as RISC. The name originally stood for Acorn RISC Machines, then later changed to Advanced RISC Machines as the market expanded beyond its original creator.

RISC itself stands for 'reduced instruction set computer', and is an idea that became popular in the '80s and '90s. This was a period when Intel and other chip makers were building more and more features and functions into the silicon, enabling programmers to execute complex operations with just a few lines of code. These processors came to be called CISC chips, for 'complex instruction set computer'.

The RISC philosophy takes the opposite approach, aiming to make a CPU as simple as possible, by reducing it to a bare minimum of basic functions. Thus the ARM architecture uses just 34 instructions, which mostly handle simple mathematical operations and move data between registers and memory locations. By contrast, the Intel 8086 supported 81 instructions, permitting far more advanced data operations – and with subsequent revisions and extensions, it's ballooned to more than 200 instructions.

The RISC approach may seem counterintuitive. The smaller instruction set means that programs need to be longer and more complex to achieve the same results. However, a RISC chip can have a much simpler physical design than a CISC one. This can make it easier and cheaper to manufacture, and it can tear through instructions at a faster rate – in most cases, every operation is completed in a single clock cycle. It can consume less power, too, which is why ARM processors are dominant in mobile phones and tablets, where battery life is paramount.

While the CISC and RISC approaches are somewhat opposed, the differences aren't as important as might be imagined. Few programs are written in pure assembly language these days, so developers don't need to worry about the underlying architecture; they can write in Python, C# or whatever language they choose and let the interpreter or compiler deal with the translation. In fact, Apple's ARM-based Macs include a real-time translation layer that lets them run programs written for x86 systems with no modifications.

The differences in power consumption are also smaller than they used to be. For many years, Intel struggled to match the low power consumption of ARM chips, not only because of the complexity of its processor designs, but also because its in-house manufacturing facilities were unable to reduce the size of the transistors inside its chips as fast as its rivals. That's been a point of some embarrassment – the very latest Intel chips are still using a 10nm

Image credit: Apple, Intel, AMD

fabrication process (dubbed Intel 7), while Apple's M-series processors have been using a 5nm process since their launch in 2020.

To help out, Intel's 12th-generation Alder Lake processors, released at the end of 2021, introduced a heterogeneous core design. Where previous Intel chips typically featured four or eight identical cores, current models combine lightweight efficiency cores (E-cores) with powerful performance cores (P-cores) that only roar into life when they're needed for the most demanding tasks. This idea was actually pioneered by ARM – it introduced what it called the big.LITTLE design in 2011 – but now that Intel has got on board, we're frequently seeing Windows laptops that can provide more than 10 hours of continuous video playback. Even so, few Windows machines can match the nigh-on 20 hours we saw from the M2 Pro-powered MacBook Pro 16-inch.

Who makes what?

Aside from the architectures, another notable difference between the two major computing architectures is that, unlike Intel, Arm doesn't make any processors of its own. Rather, the company licenses its designs to companies that can then customise them as desired, and have them manufactured to their own specification. In the case of Apple Silicon, Apple uses the core ARM logic but adds many of its own optimisations, and outsources the actual manufacturing to TSMC.

The way chips are marketed to end users is different, too. While all of Intel's x86 processors use the same underlying architecture, it's offered in an enormous number of different configurations. Within each generation of Core CPUs there are Core i3, i5, i7 and i9 variants, which further subdivide into ranges of different models aimed at mobile, desktop or gaming systems. They all have different numbers of processing cores, different amounts of cache memory, different clock speeds and different power requirements. It's confusing, and when you're choosing a computer, there's a risk that you could choose a model that's underpowered for your needs.

Apple, by contrast, offers only four chip designations per M range ie the M1, M1 Pro, M1 Max and M1 Ultra. It's a much simpler line-up than Intel's,

HIGH-PERFORMANCE COMPUTING

We hardly have to feel bad for Arm, it's not like it doesn't own the entire mobile phone and tablet space. In fact, if things weren't already looking a bit grim for Intel on the desktop and laptop front, consider that its cash cow of the server market is now under attack by not just a revitalised AMD but by Arm as well. For decades, server farms have grappled with the issue that cooling costs more than running the damn servers – two to one back in the day, even now one to one costs are a mark of a good cooling design. So, if someone appears, offering a way of reducing the power consumption of your servers, you're going to pay attention. That's exactly what Arm has been busy doing.

You might have heard of Amazon and its Amazon Web Services that run half the web. With so much bare metal to pay for, anything that saves energy benefits Amazon. So, it designed its own server processor core called Graviton, and in May 2022 it released Graviton3. It's a 64-core ARMv8.4 SoC with 64MB of L3 cache, running at 2.6GHz with 8-channel DDR5 memory and PCIe 5 on the 5nm TSMC process. Serious stuff. Estimates put power use at 100W versus 300W for a similarly pegged Xeon.

If that wasn't enough, the all-new Japanese Fugaku system has taken the top spot for supercomputing power, besting the PowerPC/Tesla-powered Summit system by 2.8x with a whopping 415.5 petaflops, using Fujitsu's 48-core A64FX Arm-based SoC. But can it play *Crysis?* The Met Office is also planning to go Arm with a supercomputer offering 145 petaflops that should claim the number three spot – enough to predict the weather before it happens!

We don't even know what you'd do with 415 petaflops – apparently it runs Linux.

Image credit: Riken

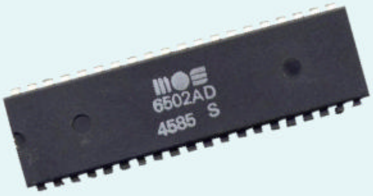
and even the regular M1 is competitive with a mid-range Intel chip; we found the original M1-powered MacBook Air achieved a multicore *Geekbench 5* score of 7,581, while the brand-new Surface Laptop 5, equipped with an Intel Core i7-1255U, turns in an almost identical 7,522. If you're comparing a range of systems, that's generally an advantage for Apple.

There's one last difference between Apple's chips and Intel's – and this one isn't intrinsic to the ARM architecture, but is a design decision that Apple has taken itself. Where Intel's chips rely on external system RAM, Apple incorporates the memory directly into the silicon die for its M-series processors.

All aboard the memory bus

This means you can't ever upgrade the memory on an Apple Silicon computer, which can lead to some agonising decisions when it comes to choosing a

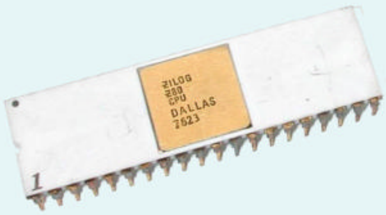
» GONE BUT NOT FORGOTTEN



MOS 6502

Founded in 1974 by former Motorola employees, MOS Technology set out to create a new processor architecture that would out-compete Motorola's chips for a much lower price. With its focus on value, the 6502 became one of the most popular processors of the early home computing revolution.

Used in: Apple II (1977), Atari 2600 (1977), BBC Micro (1981), Commodore 64 (1982), Nintendo Entertainment System (1983)



Zilog Z80

Similar to the MOS 6502, the Z80 was created in the mid-'70s by a breakaway group of chip designers – in this case, former Intel engineers. Although it could run programs written for the Intel 8080, the Z80 added many enhancements that made it more powerful and easier to program, making it attractive for personal computers and video games.

Used in: Sinclair ZX Spectrum (1982), Amstrad CPC (1984), Amstrad PCW (1985), Sega Master System (1985)

Motorola 68000

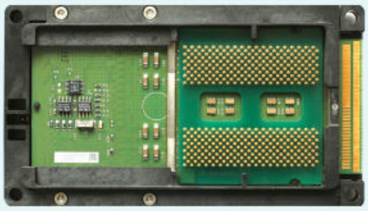
With Intel's barnstorming 8086 CPU powering the IBM PC, rival Motorola knew it had to aim high to compete. It set out to create the most powerful processor on the market, creating a new 16-bit chip with cutting-edge 32-bit features. The 68000 was initially used in high-end workstations, but as costs fell, it became the foundation of a new generation of home computers.

Used in: Apple Macintosh (1984), Commodore Amiga (1985), Atari ST (1985)

PowerPC

Jointly designed in 1992 by Apple, IBM and Motorola, the PowerPC architecture was supposed to usher in a new era of RISC computing. Apple moved its whole Mac platform from Motorola chips to the PowerPC architecture, and the architecture was adopted by several games consoles. In the end, the PowerPC architecture couldn't keep up with Intel's performance, but it's still used for some roles, under the new name of Power ISA.

Used in: Apple PowerMac (from 1992), Nintendo GameCube (2001), Microsoft Xbox 360 (2005), Sony PlayStation 3 (2006), Mars Rover Curiosity (2011)



IA-64

Created by Intel and HP in 2001, IA-64 was a 64-bit RISC platform designed to provide enterprise-class performance and scalability. Between 2001 and 2017, Intel launched 10 generations of Itanium processors based on IA-64, but system builders favoured the backwards-compatible x86-64 extensions innovated by AMD, and the Itanium line is now discontinued.

Used in: Dell Precision Workstation 730 (2001), HP ZX6000 workstations (from 2001), HPE Integrity servers (from 2001)

Image credit: Mos, Zilog, Motorola, Power pc, Intel/HP, AMD, Apple

specification. It also means that really large allocations of memory aren't available at all on the mainstream chips – the M1 is offered with a maximum of 16GB of RAM, while the M2 is limited to 24GB. If you want 32GB or more you need to move up to an expensive M2 Pro or Max system (or the M1 Ultra). For comparison, all of Intel's 12th and 13th-generation processors can use up to 128GB of RAM.

However, because Apple's RAM is literally located right next to the processor logic, and connected to it via the fastest possible fabric, its processors can access code and data extremely quickly and efficiently. The standard M1 boasts a maximum memory bandwidth of 68GB/s, while the M2 goes up to 100GB/s, and the recently launched M2 Pro and Max models go up to 200GB/s and 400GB/s respectively. With Intel it all depends on the specifics of the processor, the RAM and the motherboard, but even the newest, fastest Core i9 is limited to a theoretical maximum of 90GB/s.

What's more, Apple uses what it calls a "unified memory architecture", which means that the whole range of memory can be directly accessed by either the CPU or the on-die GPU. This provides huge efficiency benefits compared to a conventional PC architecture, where the CPU and GPU each have separate memory banks, and can't work together on the same data without copying it back and forth. It's not just games that benefit, but any process that can take advantage of GPU functions, such as video editing and rendering.

What about AMD?

So far we've been comparing Apple to Intel, but of course there's a third major player in the CPU market. AMD's chips don't have such a distinct identity, however, because they use the same core x86 architecture and instruction set as Intel.

You may wonder why Intel allows its biggest rival to use its proprietary architecture. The answer dates back to the early '80s – IBM was interested in using Intel's chips in the original IBM PC, but it didn't want to be solely reliant on one source of silicon, so it told Intel that it would use x86 processors only if a second company was also permitted to manufacture hardware under licence. AMD was thus authorised to build Intel 8086, 80186 and 80286 processors.

Later, AMD drew on this expertise to start creating its own chip designs to rival Intel's. The K5 and K6, released in the late '90s, provided x86 compatibility at a lower price than Intel's Pentium processors. And at the turn of the millennium, AMD grafted a whole new 64-bit processing mode on to the x86 architecture, with enhancements to support working with larger numbers, bigger data sets and more RAM. These extensions were then licensed by Intel, leading to a situation today where the two companies are effectively reliant on each other – Intel lets AMD use the original x86 architecture, while AMD lets Intel use the 64-bit extensions. It's hard to see either company choosing to end the arrangement.

What makes AMD better than Intel?

Although AMD's processors can run the same programs as Intel's, there are some key differences – for instance, although AMD sells its own chips, it doesn't manufacture them itself; this means that it can use whichever foundry offers the best technology. While the first two generations of Ryzen CPUs were produced by Global Foundries, AMD subsequently switched to TSMC in 2019 to take advantage of its 7nm fabrication process, and the latest Ryzen 7000-series chips use the company's 5nm process. That helps AMD chips spend more of their time running at the highest frequencies, before they need to slow down and cool off.

From left to right: AMD tends to pack in more performance cores than Intel.

Apple's M2 processors offer outstanding power efficiency.

AMD's designs also frequently pack in more performance cores than similarly priced Intel chips, in part due to its chiplet approach. Rather than build everything on to one die, it breaks the design down into multiple processor cores (chiplets) that it then connects together, along with shared resources such as the main memory cache.

> **PULLING ON A THREAD**
> "Core counts can be misleading, as both companies use multithreading technologies that allow a single core to service two execution threads at once."

Actual core counts can be misleading, because both companies use multithreading technologies that allow a single core to service two execution threads at once. Matters are further confused by Intel's recent adoption of efficiency cores, which don't contribute to peak performance.

Overall, though, you normally get more multicore processing power from an AMD chip – and to support those cores, AMD tends to provide more on-chip memory than Intel. While Ryzen CPUs don't put the whole RAM allocation on the chip die like Apple's chips, they generally have large caches that help them keep processing data and instructions at full speed, without having to wait for data to be fetched from the DIMMs.

The only question is how valuable multicore performance is. Big database servers and graphics rendering programs may benefit hugely from parallel processing power, but many games and desktop apps are mostly single-threaded. In practice, you might get a better experience with fewer, faster cores. LXF

HotPicks

Vorta » Mullvad Browser » Wike » CryFS » rmLint » Imaginer » Letterpress » cli-chess » OpenRCT2 » Boost Changer » PDF Mix Tool

Mayank Sharma might have eyesight that is not what it was, but he still has his eye on the ball when it comes to picking the best open source software out there.

BACKUP TOOL

Vorta

Version: 0.8.12

Web: https://vorta.borgbase.com

BorgBackup is one of the most powerful and feature-complete backup tools there is. However, being a CLI tool, it comes with a learning curve. If CLI isn't your thing, you can use *Vorta* to utilise the power of *Borg* through a graphical interface, and create encrypted, deduplicated and compressed backups without much trouble.

Vorta is available on Flathub and can be installed with `flatpak install flathub com.borgbase.Vorta`.

The app might not seem very friendly when you launch it for the first time, but it's actually pretty straightforward once you spend some time with it.

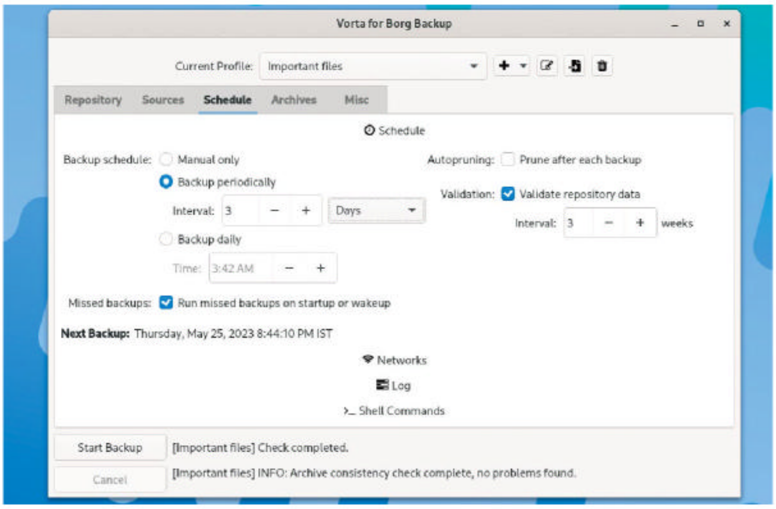
One of the best things about *Vorta* is that it enables you to create various backup profiles. These make it a lot easier for you to back up different sources to different destinations.

Talking of destinations, you can use *Vorta* to back up not only to a local storage device, but also to a remote server, or on multiple remote servers, using the same set of SSH keys, in one seamless operation. The app can back up data to local drives and external drives, as well as to remote servers including Borg's own BorgBase.

Vorta includes a configurable SSH key generator, and also has the ability to define filename or path exclusion patterns. Once you've taken your backups, *Vorta* can also help you extract, mount, delete or prune any backups.

To take a backup, head to the Repository tab and use the + button to point the app to the folder where you want to house the backups. This pops up a new window asking you to choose the location for the backup repository, along with a passphrase and encryption type.

Then head to the Sources tab and use the + button to add the files and folders you want to back up. You can also specify the aforementioned exclude patterns here to ignore backing up certain files and folders. Once everything's in order, hit the Start Backup button to take your first backup.



Vorta for Borg Backup

Current Profile: Important files

Backup schedule: ○ Manual only ● Backup periodically Interval: 3 Days ○ Backup daily Time: 3:42 AM

Autopruning: □ Prune after each backup

Validation: ☑ Validate repository data Interval: 3 weeks

Missed backups: ☑ Run missed backups on startup or wakeup

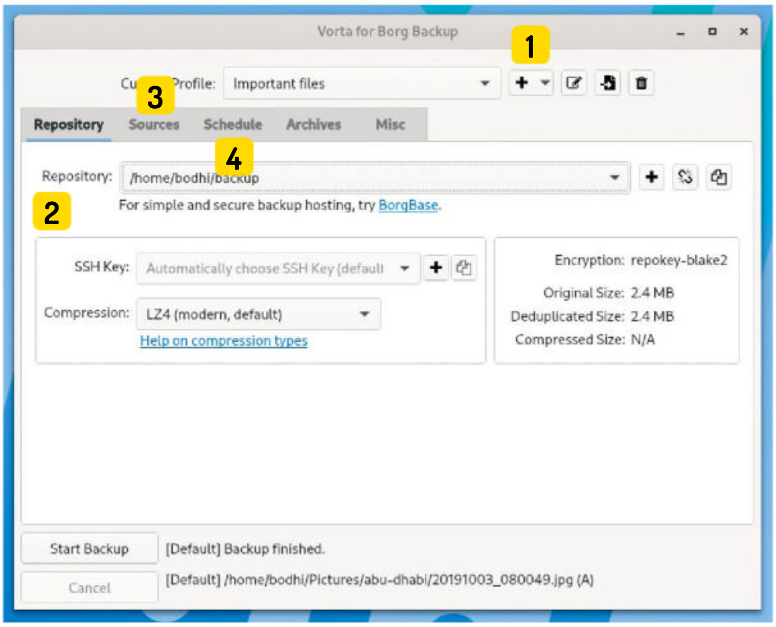
Next Backup: Thursday, May 25, 2023 8:44:10 PM IST

[Important files] Check completed.

[Important files] INFO: Archive consistency check complete, no problems found.

The app also has a comprehensive scheduler that you can use to define schedules for automatic backups.

LET'S EXPLORE VORTA...



Vorta for Borg Backup

Current Profile: Important files

Repository: /home/bodhi/backup

For simple and secure backup hosting, try BorgBase.

SSH Key: Automatically choose SSH Key (default

Compression: LZ4 (modern, default)

Encryption: repokey-blake2

Original Size: 2.4 MB

Deduplicated Size: 2.4 MB

Compressed Size: N/A

[Default] Backup finished.

[Default] /home/bodhi/Pictures/abu-dhabi/20191003_080049.jpg (A)

1 Add and select profiles Use the + icon here to create different backup profiles, which then appear in the pull-down list.

2 Configure repository You can either point the app to an existing repository or create a brand new one to get started.

3 Select sources Switch to the Sources tab and add the files and folders you want to back up.

4 Define a schedule The Start Backup button takes backups as soon as you click on it. You can switch to the Schedule tab to define a schedule for automatic backups.

WEB BROWSER

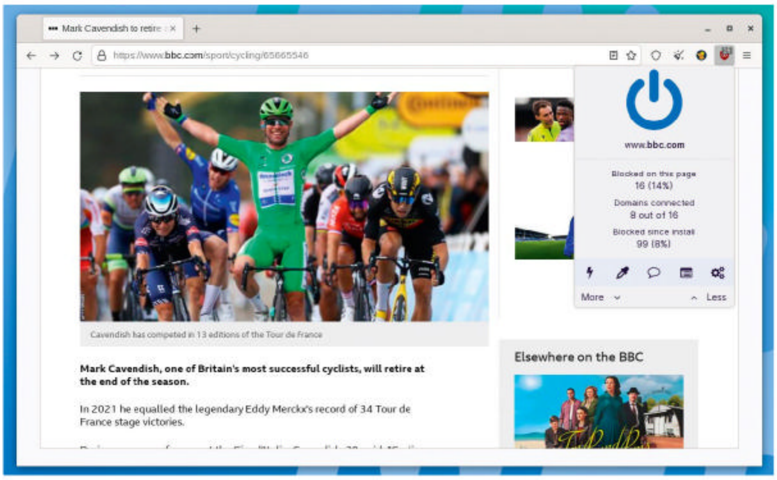
Mullvad Browser

Version: 12.0.4
Web: https://mullvad.net/en/browser

We're big fans of the *Tor* browser. It's become the go-to option for the privacy-conscious thanks to its ability to conceal the user's identity, physical and digital, by rerouting connections to volunteer-run encrypted servers across the world.

Tor's developers have collaborated with the Mullvad VPN service to create a privacy-centric browser that doesn't use the Tor network, yet extends the privacy advantage of the *Tor* browser. To install the *Mullvad* browser, download the latest version and extract it with `tar xf mullvad-browser-linux64-12.0.4_ALL.tar.xz`. Head to the extracted directory and fire up the browser with `./start-mullvad-browser.desktop`.

Mullvad appears like a normal web browser, and you use it as such as well. By default, it comes with private mode enabled, which means it won't preserve cookies between sessions. It also discards the list of visited pages, search bar entries and any cached web content.



Cavendish has competed in 13 editions of the Tour de France

Mark Cavendish, one of Britain's most successful cyclists, will retire at the end of the season.

In 2021 he equalled the legendary Eddy Merckx's record of 34 Tour de France stage victories.

Interestingly, *Mullvad* isolates cookies in separate cookie jars so trackers cannot connect to each other, preventing them building a digital persona of the user.

Furthermore, the browser also helps you hide personal data and online activity by concealing your metadata. It also assigns all its users with a similar digital fingerprint by hiding clues like screen resolution, WebGL and common APIs, and restricting fonts.

To further reduce your digital footprint, *Mullvad* integrates only a few vital plugins, such as the popular ad-blocker *uBlock Origin*. One of its most interesting features is dubbed New identity, which essentially zaps all cookies immediately without closing the browser, making it virtually impossible for trackers to track you.

You don't have to be a Mullvad VPN user to use the Mullvad browser, although its developers recommend using it together with a VPN service for further privacy.

WIKIPEDIA READER

Wike

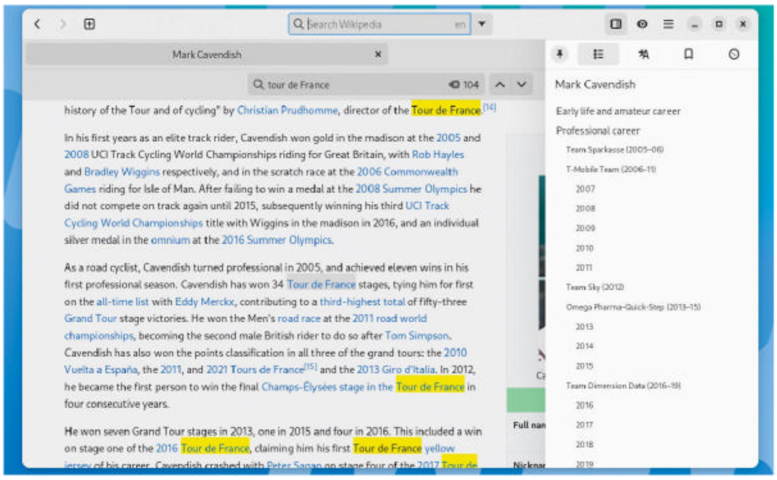
Version: 2.0.1
Web: https://hugolabe.github.io/Wike/

If you love Wikipedia, you can use *Wike* to access it from the comfort of a distraction-free desktop app, along with extra features and better integration than you get with a web browser.

The app is available on Flathub, and can be installed with `flatpak install flathub com.github.hugolabe.Wike`. Once installed, you can launch it from the distro's Applications menu or with `flatpak run com.github.hugolabe.Wike`.

Wike has an intuitive interface, which also very helpfully offers several colour schemes such as Sepia, Dark and Light. Furthermore, the app also enables you to customise various aspects of the interface, including the spacing and fonts, to make the reading experience more comforting.

Articles within *Wike* work and behave pretty much as they do on the Wikipedia website. Clicking on a link in an article takes you to it within the app. Furthermore, you can also enable link previews, after which you're able to hover over internal links to get an excerpt and an image preview of the link.



history of the Tour and of cycling" by Christian Prudhomme, director of the Tour de France.[14]

In his first years as an elite track rider, Cavendish won gold in the madison at the 2005 and 2008 UCI Track Cycling World Championships riding for Great Britain, with Rob Hayles and Bradley Wiggins respectively, and in the scratch race at the 2006 Commonwealth Games riding for Isle of Man. After failing to win a medal at the 2008 Summer Olympics he did not compete on track again until 2015, subsequently winning his third UCI Track Cycling World Championships title with Wiggins in the madison in 2016, and an individual silver medal in the omnium at the 2016 Summer Olympics.

As a road cyclist, Cavendish turned professional in 2005, and achieved eleven wins in his first professional season. Cavendish has won 34 Tour de France stages, tying him for first on the all-time list with Eddy Merckx, contributing to a third-highest total of fifty-three Grand Tour stage victories. He won the Men's road race at the 2011 road world championships, becoming the second male British rider to do so after Tom Simpson. Cavendish has also won the points classification in all three of the grand tours: the 2010 Vuelta a España, the 2011, and 2021 Tours de France[15] and the 2013 Giro d'Italia. In 2012, he became the first person to win the final Champs-Élysées stage in the Tour de France in four consecutive years.

He won seven Grand Tour stages in 2013, one in 2015 and four in 2016. This included a win on stage one of the 2016 Tour de France, claiming him his first Tour de France yellow jersey of his career. Cavendish crashed with Peter Sagan on stage four of the 2017 Tour de

Mark Cavendish
Early life and amateur career
Professional career
Team Sparkasse (2005–06)
T-Mobile Team (2006–11)
2007
2008
2009
2010
2011
Team Sky (2012)
Omega Pharma-Quick-Step (2013–15)
2013
2014
2015
Team Dimension Data (2016–19)
2016
2017
2018
2019

Wike has a multi-tabbed interface and supports over 300 languages. Some of the features that make it more useful than browsing Wikipedia with a web browser is its ability to bookmark articles in custom lists, as well as the history of recent articles. *Wike* also has impressive search abilities and provides relevant suggestions to help you find new content.

We also like the custom Table Of Contents side panel that helps you quickly scroll through the various sections of an article. From here, you can also access links to the same article in other languages, and more.

On the downside, the app lacks the ability to cache articles for offline reading. However, it does let you export articles as a PDF via its Print options.

Wike has impressive search abilities and besides Wikipedia can also search within the articles.

FILE ENCRYPTION

CryFS

Version: 0.11.3
Web: www.cryfs.org

CryFS is an encryption tool for safeguarding all files, especially in the cloud. It's easy to set up, runs in the background, and works well with popular cloud services such as Dropbox and OneDrive.

It is available in the official repos of Debian and Ubuntu distros, and can be installed with `sudo apt install cryfs`. Users of other distros can install it via *HomeBrew (HotPicks,* **LXF303***)* with `brew install cryfs`.

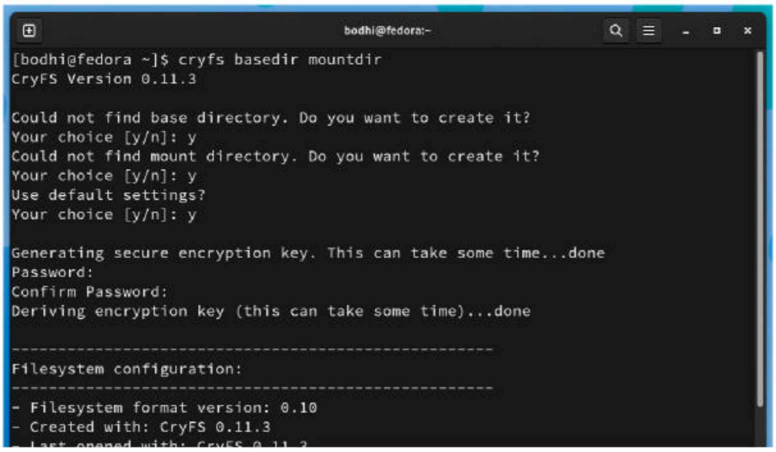
Once installed, you can create an encrypted directory with something like `cryfs basedir mountdir`. The command asks you some questions about the configuration of your encrypted directory. You can safely go with the default options. You're also prompted for an encryption password.

Once the directories have been mounted, you can modify the content inside the **mountdir** directory. *CryFS* then encrypts them and stores the encrypted data in the **basedir** directory. If the corresponding directories don't exist yet, *CryFS* offers to create them.

The unencrypted content of **mountdir** will never be stored to the hard disk, so everything on your hard disk is encrypted. To unmount the encrypted directory, specify the mount directory (**mountdir** in this case) type, such as `cryfs-unmount mountdir`.

When *CryFS* isn't running, you won't be able to access your encrypted data, and the contents of **mountdir** will be empty. On the other hand, while the **basedir** won't be empty, its contents will be encrypted and cannot be read.

Although *CryFS* can also be used to just encrypt files locally, its main purpose is to be used together with cloud storage providers such as Dropbox. It should work well together with any cloud storage provider that uses a local synchronisation folder. To upload the encrypted files, make sure you put **basedir** inside the folder that's synchronised with the cloud service.



```
[bodhi@fedora ~]$ cryfs basedir mountdir
CryFS Version 0.11.3

Could not find base directory. Do you want to create it?
Your choice [y/n]: y
Could not find mount directory. Do you want to create it?
Your choice [y/n]: y
Use default settings?
Your choice [y/n]: y

Generating secure encryption key. This can take some time...done
Password:
Confirm Password:
Deriving encryption key (this can take some time)...done

----------------------------------------------------
Filesystem configuration:
----------------------------------------------------
- Filesystem format version: 0.10
- Created with: CryFS 0.11.3
- Last opened with: CryFS 0.11.3
```

In addition to the file content itself, CryFS ensures that no data, including directory structure, and metadata, leaves your computer unencrypted.

DISK CLEANER

rmLint

Version: 2.10.1
Web: http://rmlint.rtfd.org

If you're like us, the only thing that collects more lint than the washing machine is your distro. *RmLint* is a CLI tool that can traverse the filesystem and identify all kinds of digital lint clogging up the installation, including duplicate files and directories, broken symbolic links and more.

Many major Linux distros already package *rmlint*. On Debian-based distros like Ubuntu use `sudo apt install rmlint`, while on Fedora use `sudo dnf install rmlint`. To put it to use, simply type `rmlint` in the terminal.

This scans the current working directory, identifies the lint, and lists it all in the terminal. It also prints a summary of the total files scanned and the size of the duplicate files, along with other details.

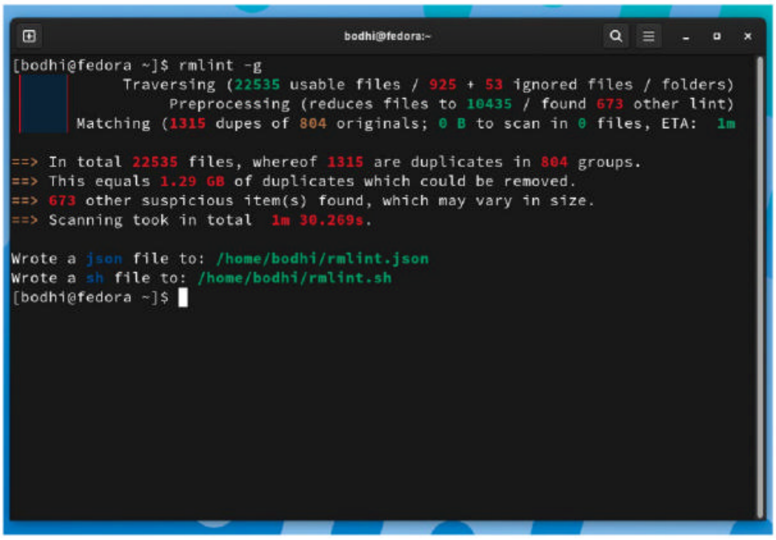
Note that the command doesn't delete these files per se, but rather generates executable output, such as JSON or shell scripts that you can use to delete the files.

By default, the command writes a shell script named **rmlint.sh** that contains the shell commands to remove duplicates and other lint. Before running the script, it's always advised to first review the content of the **rmlint.sh** script (by opening it in a text editor) to check what it plans to delete and make any edits as necessary.

When sure it's good to go, let it rip with `./rmlint.sh`. The script asks for confirmation, then deletes the lint, before self-destructing, *Mission Impossible*-style.

Like all good CLI tools, *rmlint* has a good repertoire of options, and it's a good idea to spend some time reading through its man page to explore them.

For instance, you can ask it to only check files between 500MB and 2GB with `rmlint --size 500M-2G`. Similarly, you can ask it to only check files that are larger than 10MB with `rmlint -s 10M`.



```
[bodhi@fedora ~]$ rmlint -g
            Traversing (22535 usable files / 925 + 53 ignored files / folders)
                 Preprocessing (reduces files to 10435 / found 673 other lint)
       Matching (1315 dupes of 804 originals; 0 B to scan in 0 files, ETA:  1m

==> In total 22535 files, whereof 1315 are duplicates in 804 groups.
==> This equals 1.29 GB of duplicates which could be removed.
==> 673 other suspicious item(s) found, which may vary in size.
==> Scanning took in total  1m 30.269s.

Wrote a json file to: /home/bodhi/rmlint.json
Wrote a sh file to: /home/bodhi/rmlint.sh
[bodhi@fedora ~]$ 
```

When scanning a large directory, instead of a scrolling list of files, you can ask rmlint to instead show a progress bar with the -g switch, such as "rmlint -g".

AI IMAGE GENERATOR

Imaginer

Version: 0.2.0
Web: https://imaginer.codeberg.page

AI image generation is all the rage these days. Using *Imaginer,* you can call upon the power of image-generation AI algorithms to cobble up some creative artwork right from your desktop.

It's available in the repos of some distros, but it's best to grab the distro-agnostic version from Flathub with `flatpak install page.codeberg.Imaginer.Imaginer`.

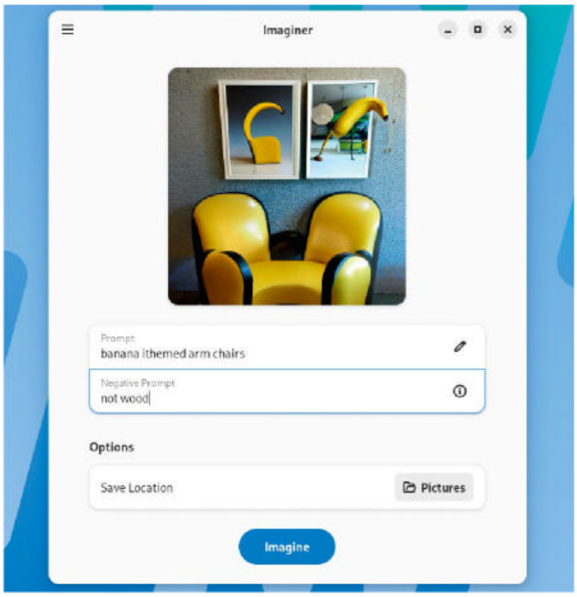
The app has a straightforward interface. To call on the services of the AI maestro, enter keywords or text prompts in the Prompt field, and hit the Imagine button. The app ferries the prompt to the AI image-generation service, which works its magic and produces an image.

You can refine things by adding a negative prompt. This extra set of keywords informs the AI image-generation engine about details not to generate. For instance, type `low quality` in the negative prompt to ensure the engine only generates high-quality artwork.

You can also ask the app to automatically save any generated imagery, specifying a location in the Save Location section. It's a good idea to define the location before you generate an image because the app offers no mechanism for manually saving an image after it has been generated.

One of the best aspects of *Imaginer* is that it's not limited to just one AI image-generation engine. You can choose from a plethora of services including Open AI, Stable Diffusion, Analog Diffusion, Anything, Portrait Plus, Nitro Diffusion, Open Journey and Waifu Diffusion.

Head to Preferences from the hamburger menu to enable the service you want to use. Once enabled, you can select the service from the Providers list.



Prompt: banana ithemed arm chairs

Negative Prompt: not wood

Save Location: Pictures

Think of the prompt and the negative prompt as the dos and don'ts for image generation.

ASCII IMAGES

Letterpress

Version: 1.3.0 **Web:** https://gitlab.com/gregorni/Letterpress

ASCII art has lost its sheen, and its popularity isn't what it used to be back in the days of dialup internet and BBS (*alright granddad!–Ed*) but it still has fans. As ASCII aficionados will recall, generating ASCII images took a lot of time and effort. *Letterpress* lets you relive those glorious days, without the sweat, by turning your pictures into ASCII images.

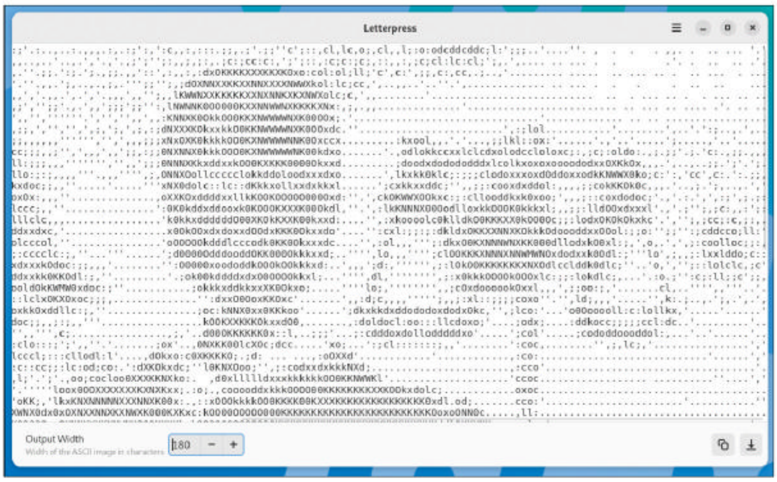
The app is available on Flathub and can be installed with `flatpak install flathub io.gitlab.gregorni.ASCIIImages`.You can then run it from the distro's Applications menu or with `flatpak run io.gitlab.gregorni.ASCIIImages`.

Letterpress gets its powers from *jp2a*, which is an ASCII converter engine. You can use it to convert any JPEG or PNG image into ASCII art.

Working with the app is pretty simple. Just load any photo from your computer into *Letterpress*, either with the Open File button and navigating the filesystem or just dragging and dropping an image into the app's interface. *Letterpress* then works its magic and spits out an ASCII version of the image in mere seconds.

Once this is done, you can choose the width of the ASCII characters from the bottom bar, before downloading the result to a TXT file. The app also offers the option to save it to the clipboard, from where you can paste it anywhere you want.

While using the app is pretty straightforward, getting a meaningful image is a lot more cumbersome. *Letterpress* works as advertised and replaces the pixels in the original image with one of the 128 ASCII characters. However, depending on the complexity of the source image, the resulting ASCII image doesn't always make sense. For some images, increasing the width of the image does the trick and makes the output visually appealing.



Output Width: 180
Width of the ASCII image in characters

Guess the image competition! Let us know what you think it is and we'll send you our complimentary compliments!

CHESS GAME

cli-chess

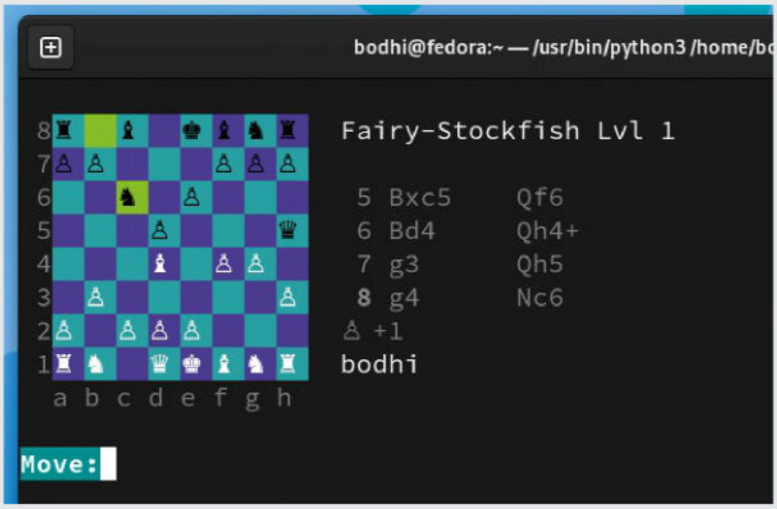
Version: 1.0.0 **Web:** https://github.com/trevorbayless/cli-chess/

This one comes courtesy of one of our readers who has created a terminal chess game called *cli-chess*. The game is developed around a chess engine that supports offline playing of not just the regular chess game, but also several Lichess variants, including Crazyhouse, King of the Hill, Three-check and more. Furthermore, you can use *cli-chess* to play with other players online on the **Lichess.org** server, and also stream games from Lichess TV.

The game is written in Python and can be installed using the *Pip* package manager with `pip install cli-chess`. Once it's done, type in `cli-chess` to begin playing. You can use the keyboard to navigate the menu and select the variant you'd like to play. After making your selection, press F1 to begin the game. By default, the game initiates the standard chess game against the computer.

Once inside the game, you can't move the pieces with the arrow keys or your mouse. Instead, you have to use either SAN (standard algebraic notation) or LAN (long algebraic notation) to make moves, such as c6 (pawn to C6), Be2 (bishop to E2) or Nxe7 (knight captures the piece on E7). To help you make the move, the game mentions the coordinates by the side of the board.

In order to play against other players online, you first need to create an API token for your Lichess account, and then pass it to the game to help it authenticate with the service. Fire up your browser and log in to your Lichess account, and then follow the link on the project's website to generate a Lichess API token for *cli-chess* to authenticate with. Finally, copy the token and pass it on to the game with `cli-chess --token <your-token>`.



```
8                          Fairy-Stockfish Lvl 1
7
6                           5 Bxc5      Qf6
5                           6 Bd4       Qh4+
4                           7 g3        Qh5
3                           8 g4        Nc6
2                          ♙ +1
1                          bodhi
  a b c d e f g h

Move:
```

You can increase the size of the terminal font in order to make the board bigger.

AMUSEMENT PARK SIMULATOR

OpenRCT2

Version: 0.4.4
Web: https://openrct2.io

Who hasn't thought of owning an amusement park? That's exactly what you can do with *OpenRCT2*. The game puts you in charge of building and maintaining your very own park.

You can easily install the game using the official AppImage atop any Linux distro with `flatpak install flathub io.openrct2.OpenRCT2`.

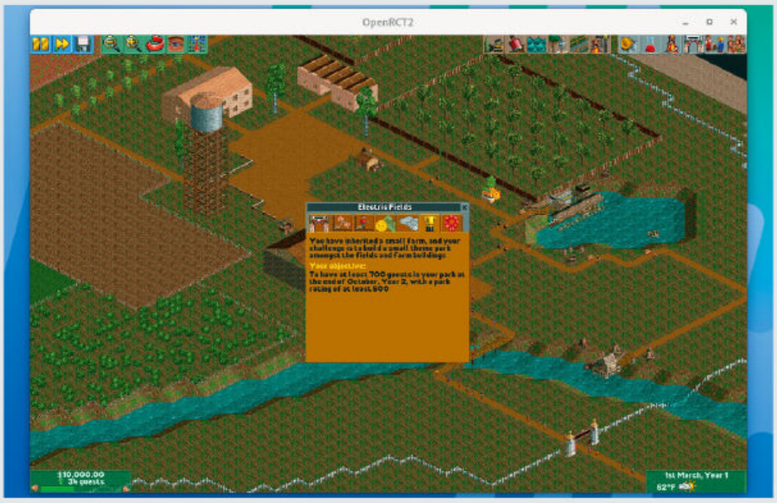
OpenRCT2 is an open source re-implementation of *RollerCoaster Tycoon 2 (RCT2)*. It attempts to provide everything from *RCT2* as well as many enhancements and additional features. It also needs files from the original *RollerCoaster Tycoon 2* (£7 on GoG) or *RollerCoaster Tycoon Classic* in order to work.

If you have a GOG version of *RCT2*, you must install *Innoextract* to extract the contents from the installer. This is available in the official repos of most distros and can be installed with `sudo dnf install innoextract` on Fedora and `sudo apt install innoextract` on Ubuntu.

Create a folder for the extracted files and move the *RCT2* installer executable inside it. Use *Innoextract* to dump the contents of the installer executable, such as `innoextract setup_rollercoaster_tycoon2_2.0.0.6.exe`. When you launch *OpenRCT2*, you're asked for the location of the *RCT2* installation, so you should point it to the contents in the extracted folder.

The game involves building and maintaining an amusement park with attractions, shops and other facilities. Your objective is to lay it out such an way so as to entertain the guests and turn a profit.

It allows for two types of gameplay modes. In the Scenario mode, players must complete a certain objective in a specified time limit, while in the Sandbox mode, you have more flexibility and can build your amusement park with no restrictions.



You have inherited a small farm, and your challenge is to build a small theme park amongst the fields and farm buildings

Your objective:

To have at least 700 guests in your park at the end of October, Year 2, with a park rating of at least 600

£10,000.00
34 guests

1st March, Year 1
62°F

OpenRCT2 isn't difficult to master, but could prove to be a little challenging for anyone who hasn't played RCT2.

CPU CONTROLLER

Boost Changer

Version: 5.0.2 **Web:** https://github.com/nbebaw/boostchanger

Boost Changer is a simple app to help you alter your processor's frequency. It initially worked only on Intel CPUs, but the latest release can also alter the frequency of AMD processors.

It is available as a distro-agnostic AppImage. You can grab the latest release from its website, then give it executable permissions with the `chmod` command, such as `chmod +x boostchanger-5.0.2.AppImage`.

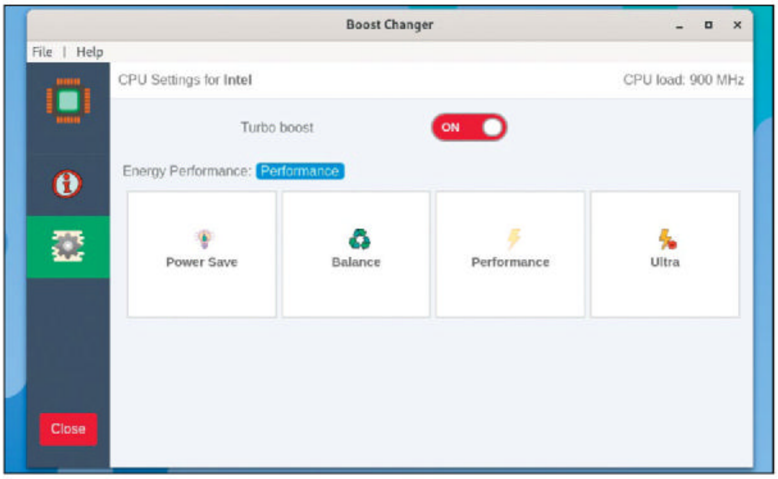
Then double-click the AppImage or fire it up from the terminal with `./boostchanger-5.0.2.AppImage`.

The app has a simple interface. The landing page is a dashboard that displays all kinds of information about your installation, such as the distro name, running kernel version and uptime, plus details about hardware including the CPU, memory, disk type, size and vendor.

The real magic happens in the CPU Settings section, which you can switch to from the navigation menu on the left. From here you can choose between four predefined self-explanatory modes: Power Save, Balance, Performance and Ultra.

Note that even though the app doesn't need root privileges to run, as soon as you select a mode, a pop-up window asks for the super user password. The Energy Performance parameter then changes to reflect the current enabled mode.

The developers point out that *Boost Changer* isn't designed to replace other battery-life-optimising utilities such as *TLP* and *PowerTOP*. Instead, the app can be thought of as a graphical front-end for quickly altering the clock speed of the CPU using the sysfs pseudo filesystem. Also note that the app currently only works on real CPUs, and not on virtual hardware.



CPU Settings for Intel

CPU load: 900 MHz

Turbo boost: ON

Energy Performance: Performance

Use Boost Changer to alter the clock speed of your CPU in order to either maximise performance or save energy.

PDF EDITOR

PDF Mix Tool

Version: 1.1.1 **Web:** https://scarpetta.eu/pdfmixtool/

If you work with PDF files, *PDF Mix Tool* is a wonderful utility to have in your repertoire. The app enables you to perform common editing operations on PDF files, such as merging two or more PDF files, changing the orientation of the pages inside the PDF, extracting pages or images from a PDF, and a whole lot more.

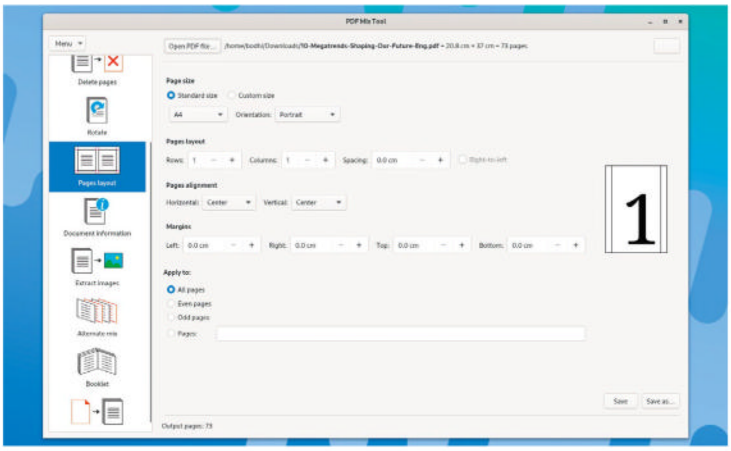
PDF Mix Tool is available in the repos of most distros. Users of Debian-based distros such as Ubuntu can fetch it with `sudo apt install pdfmixtool`, while those using Fedora and its derivatives can use `sudo dnf install pdfmixtool` to install the app. If your distro doesn't have the latest version of the app, you can also fetch it from either Flathub or the Snap Store.

The app has a fairly intuitive interface. All the supported operations are listed in the left-hand column. Begin by selecting the operation you are interested in. Depending on the selected operation, you see the relevant options on the right-hand panel.

For instance, if you select the Merge PDF Files option, the app then asks you to point it to the PDF files you want to merge. After adding in all the files you want to merge, you can rearrange their order by using the Move Up and Move Down buttons. When you're done, hit Generate PDF to create the new combined PDF.

Similarly, to extract pages from a PDF, select the Extract Pages option, point the app to the relevant PDF file, and then select the pages you want to extract. You can save the extracted pages as individual PDF files or as a single PDF. The Delete Pages option works in the same fashion, but instead zaps the specified pages.

Another interesting option is the Pages Layout section, which you can use to alter the orientation of the pages inside a PDF.



/home/bodhi/Downloads/10-Megatrends-Shaping-Our-Future-Eng.pdf • 20.8 cm × 37 cm • 73 pages

Page size: ● Standard size ○ Custom size; A4; Orientation: Portrait

Pages layout: Rows: 1; Columns: 1; Spacing: 0.0 cm; □ Right-to-left

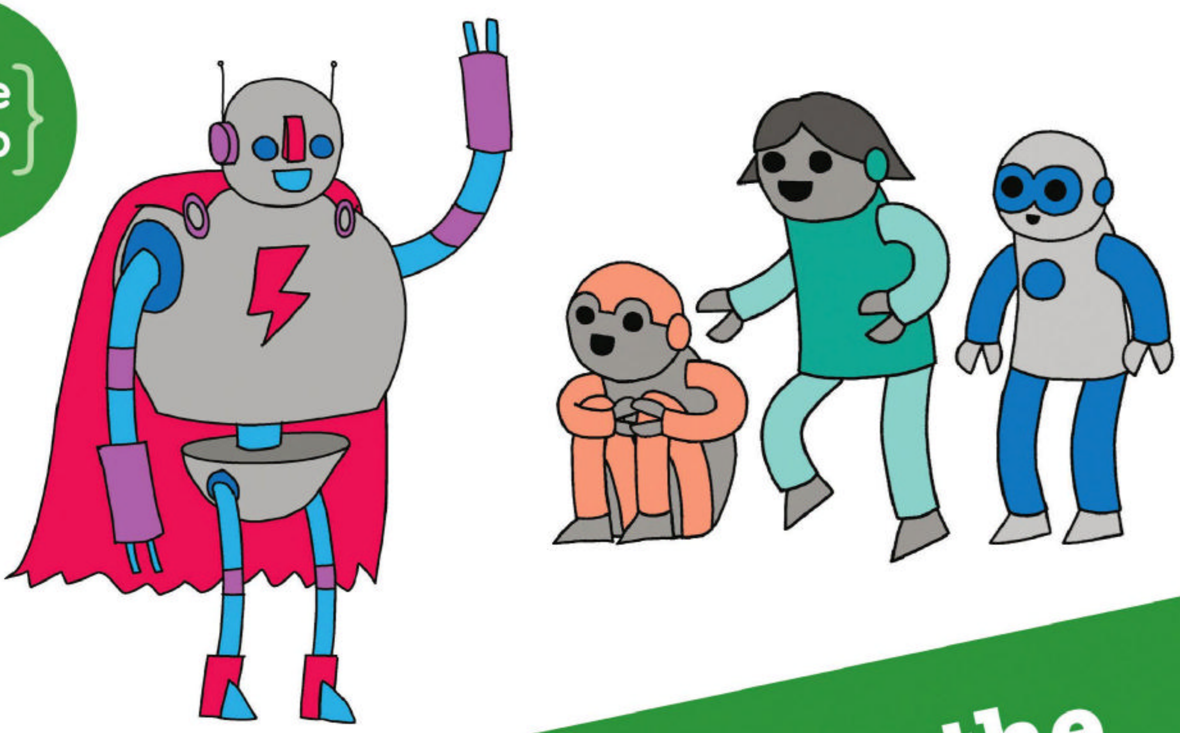
Pages alignment: Horizontal: Center; Vertical: Center

Margins: Left: 0.0 cm; Right: 0.0 cm; Top: 0.0 cm; Bottom: 0.0 cm

Apply to: ● All pages ○ Even pages ○ Odd pages ○ Pages:

Output pages: 73

In addition to altering PDFs, you can also use the app to convert a PDF file into a booklet.



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**

MU

Credit: https://codewith.mu

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

Code your own Python text adventure

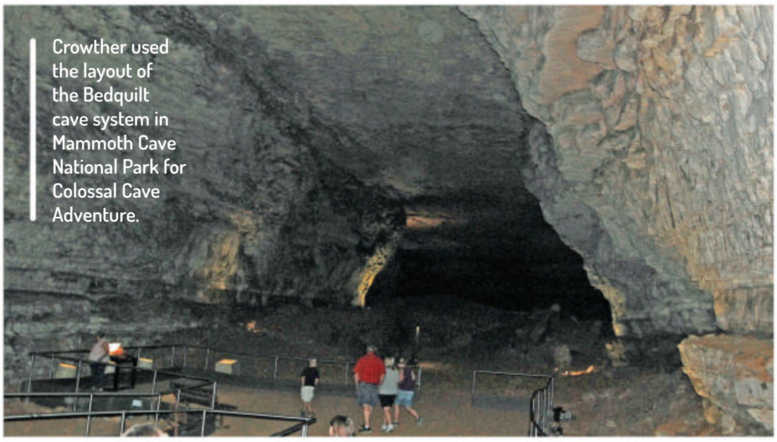
Nate Drake guides you through creating your own interactive text adventure with the power of Python, starting with the basics...

OUR EXPERT

Nate Drake wants to dedicate this series to his dad, who once told him that no one was ever going to pay him to sit around and play video games.

Those of us old enough to remember the '70s and early '80s know that games were graphically very uninspiring then. Not to mention the fact that personal computers were still prohibitively expensive.

In 1975, MIT student William Crowther took some time out of his busy schedule developing ARPANET to create a text adventure game he could play with his kids. Although it wasn't the first of its kind (that honour belongs to 1973's *Hunt the Wumpus), Colossal Cave Adventure* quickly spread over the early internet after heavy modifications.



Crowther used the layout of the Bedquilt cave system in Mammoth Cave National Park for Colossal Cave Adventure.

CREDIT: Jud McCranie, CC BY-SA 4.0, https://commons.m.wikimedia.org/wiki/File:Mammoth_cave,_Kentucky,_US,_tourists_inside.jpg

The basic premise was the same as any text adventure, in that you entered commands, such as the cardinal points of the compass (N, S, E and W), to move between areas, and interact with items, traps and monsters in a bid to win through to the end with maximum points.

Younger readers may find it difficult to believe that people actually still do this for fun. In fact, IF (interactive fiction) has never been more popular. The annual XYZZY Awards even offer prizes to those who code the very best text adventure games.

In this four-part series, we plan to take you through the basic steps of creating your own text adventure game using the Python 3 programming language. The good news is that since it's text-based, you won't need a degree in *Photoshop* wizardry. It's also a great programming project for getting started with Python, as well as developing clean coding practices.

If you're a Linux user, there's more good news, because support for Python 3 is almost certainly already built into your distro. You can just type `python3` via the terminal to get started.

Still, this isn't the easiest way to code and run Python scripts, so we've used the *Mu* Python interpreter instead. It's available in the Ubuntu software centre and it offers a quick and easy way to run scripts. *Mu's* also available for the Raspberry Pi.

Starting with how not to do it...

In the first part of this guide, we're going to focus on getting set up and world-building. In other words, you will learn how to create rooms in your underground cavern, dungeon or mystical world, as well as navigate between them.

It's very easy to imagine that this is a lot easier than it is. For instance, try pasting this script into your Python interpreter and running it:

```
print("Which way will you go?")
action_input = input('Action: ')
if action_input == 'n':
    print("To the North, you see the skeleton of a dwarf holding an axe.")
elif action_input == 's':
    print("To the South, you see an antique wooden chest. It's locked.")
elif action_input == 'e':
    print("You see entrance to the cave where you arrived earlier. The gate is locked.")
elif action_input == 'w':
```

QUICK TIP

To download all the code samples for part one of this series, open a terminal and install Git via `sudo apt-get install git`**. Next run** `git clone https://github.com/azuregate/lxfpython textadventure.git`**. The files download to the 'lxfpython adventure' folder in your home directory.**

```
 print("To the west you see an iron grille set into the wall.")
else:
 print("You cannot go that way.")
```

If you're using *Mu,* remember that you need to save code with a valid filename, such as **test1.py**, before you can run the script.

In order to break this code down, the `print` command is being put to fine use here. Setting the variable `action_input` then matches any of the lower-case letters `n`, `s`, `e` or `w` so that the script prints out a description of the new room.

The `elif` command is Python's shortened version of `ELSE IF`, which simply means if the input doesn't match, to carry out another query. If you don't input a valid direction, this script just tells you, 'You cannot go that way.'

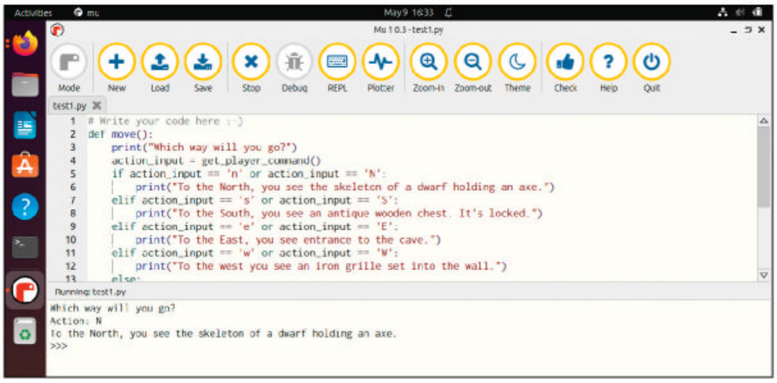
While you can code an adventure in this way, it will cause headaches later on. Firstly, you are probably going to want to have more than five rooms. This means you'll have to include more lines of `print` and `elif` commands to describe each one. Most text adventures also allow you to interact with the room in some way. What if you want to pick up the axe or try to open the gate?

Since you're probably going to want to do other things besides drift between rooms in future, it's best to define movement as a specific function. You can create functions using the `def` command in Python. The syntax is quite simple, for example:

```
def hello_world():
  print("Hello World")
```

You can then call this function by just inserting `hello_world()` anywhere into your script.

As things stand, this script simply exits gracefully as soon as you've moved to one room. Luckily, Python lets you use the `while` command to loop code, which you can insert into your new `move` function:



```
# Write your code here :-)
def move():
    print("Which way will you go?")
    action_input = get_player_command()
    if action_input == 'n' or action_input == 'N':
        print("To the North, you see the skeleton of a dwarf holding an axe.")
    elif action_input == 's' or action_input == 'S':
        print("To the South, you see an antique wooden chest. It's locked.")
    elif action_input == 'e' or action_input == 'E':
        print("To the East, you see entrance to the cave.")
    elif action_input == 'w' or action_input == 'W':
        print("To the west you see an iron grille set into the wall.")
    else:
```

Running: test1.py

```
Which way will you go?
Action: N
To the North, you see the skeleton of a dwarf holding an axe.
>>>
```

Mu is an excellent Python editor. You can write and run code from within the program. Make sure to save first.

```
def move():
 print("Which way will you go?")
 while True:
  action_input = get_player_command()
   if action_input in ['n', 'N']:
     print("To the North, you see the skeleton of a dwarf holding an axe.")
   elif action_input in ['s', 'S']:
      print("To the South, you see an antique wooden chest. It's locked.")
   elif action_input in ['e', 'E']:
       print("To the East, you see entrance to the cave.")
    elif action_input in ['w', 'W']:
       print("To the west you see an iron grille set into the wall..")
    else:
       print("You cannot go that way.")
def get_player_command():
 return input('Action: ')
move()
```

This version of the code is a slight improvement. There's now a defined function for moving. Players can

MAPPING YOUR WORLD

Back in the day, budding game designers would create the maps for their text adventures using graph paper and a lot of patience. You're welcome to try this if it tickles your fancy, but there are more modern alternatives.

One of these is *IFMapper* (**https://tinyurl.com/ifmapper**). This handy tool enables you to plot individual rooms, map routes between them and even list a room description and items.

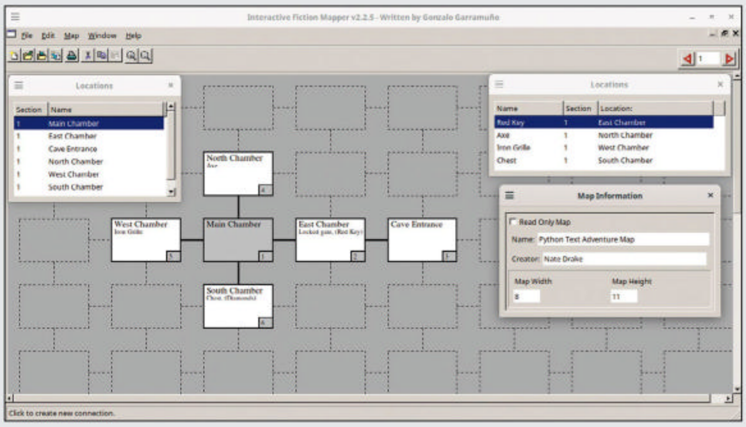
Doing this now will give you a much clearer idea of how to code the game later on. *IFMapper's* written in Ruby, so to get started, run `sudo apt-get install ruby` in the terminal.

Install the dependencies by running `sudo apt-get install build-essential libfox-1.6-dev ruby-dev libxrandr-dev` then install *IFMapper* itself with `sudo gem install ifmapper`. You can now run the program at any time by typing `IFMapper` into the terminal.

Once the program launches, go to File > Save and give your map a memorable name. Click on any box in the grid to create a New Location. Double-click it to change the name, such as Main Cavern.

You can use the other tabs to add a list of items in the room (if there are any), as well as a description. Click the dotted lines between rooms to represent a route between them. You can delete a route or room simply by clicking on it and then pressing Del.

If you need some inspiration, go to File > Open and check out the **maps** directory for some examples.



Section	Name
1	Main Chamber
1	East Chamber
1	Cave Entrance
1	North Chamber
1	West Chamber
1	South Chamber

Name	Section	Location:
Red Key	1	East Chamber
Axe	1	North Chamber
Iron Grille	1	West Chamber
Chest	1	South Chamber

□ Read Only Map
Name: Python Text Adventure Map
Creator: Nate Drake
Map Width: 8
Map Height: 11

Tools such as IFMapper enable you to easily plot out the areas and rooms in your game.

CREDIT: https://ggarra13.github.io/ifmapper/en/start.html#WhatisIFMapper

QUICK TIP

If you prefer not to install extra software, the terminal on most Linux distros can run Python perfectly well. To execute a script, just use the `python3` command plus the filename, such as `python3 adventure1.py`.

now enter directions in upper or lower-case. After you've made a move, the game asks where to go next.

A touch of class

Now that you have your movements down pat, it's time to start doing some world-building. This is where Python comes to the rescue through its use of classes and objects.

A class is best described as a kind of blueprint that allows you to define objects. A class defines the attributes a method can have. For instance, each room in your adventure will be in a specific location and have a specific description. To get started, define the **room** class by placing this code at the top of a new script.

```
class room:
 def __init__(self, number, name, description, exits):
  self.number = number
   self.name = name
   self.description = description
```

Now you've defined the **room** class, you can actually list a specific room below it, for example:

```
room1 = room(1, “Main Cavern”, “You are standing in a
dark cavern. You hear the sound of dripping water.”)
```

You can now interact with the object attributes from within your script using the format `object.attribute`. For instance, try adding these lines at the end of your code to see the name of the room and its description:

```
print (room1.name)
print (room1.description)
```

This is fine for providing a description of the very first room, but what about when you move to others? Once again, classes and objects can come to the rescue. Return to the top of your script and create a **player** class below the **room** one, as follows:

```
class player:
  def __init__(self, name, currentroom):
    self.name = name
    self.currentroom = currentroom
```

Now you can change the player's location – to **room1**, for example – and display the name and description for that room. Just add these lines to the bottom of your script and run it:

```
player.currentroom = room1
print(player.currentroom.name)
print(player.currentroom.description)
```

Try adding other rooms besides **room1**. Next try adding these lines to the end of your script:

```
Player.currentroom = room1

def move():
print (player.currentroom.name)
 print (player.currentroom.description)
print(“Which way will you go?”)
 while True:
   action_input = get_player_command()
    if action_input in [‘n’, ‘N’]:
     player.currentroom = room2
     move()
    elif action_input in [‘s’, ‘S’]:
     player.currentroom = room3
     move()
     elif action_input in [‘e’, ‘E’]:
     player.currentroom = room4
     move()
     elif action_input in [‘w’, ‘W’]:
     player.currentroom = room5
     move()
     else:
```

» KEEPER OF THE KEYS

No text adventure would be complete without having the player negotiate locked doors. To get started, first create a **key** class at the top of your script:

```
class key:
  def __init__(self, keyusedesc, keyroomdesc, keyexits):
    self.keyusedesc = keyusedesc
    self.keyroomdesc = keyroomdesc
    self.keyexits = keyexits
```

You should also update the **room** class to have an optional key attribute **self.key**. The **player** class also needs to be updated with their own **self.keyring** attribute, so the script can check if they have the right key for this room. You can now create a specific **key** object below where the exits are defined:

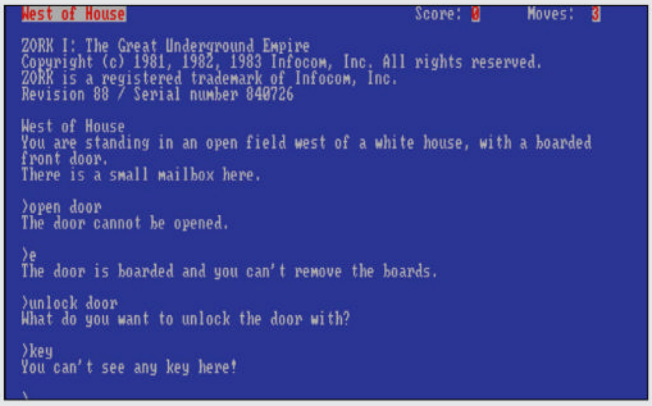
```
redkey = key(“You use the red key to unlock the gate.”, “You
see the entrance to the cave where you arrived earlier. The
red gate lies open.”, [None,None,room6,room1])
```

You can now add:

```
room4.key = redkey
player.keyring = [None,None,None,None,redkey]
```

Finally, you need a function to check whether a key's required for the current room and to use it if so:

```
def checkkeys():
  currentroom = player.currentroom
   if currentroom.key != None and currentroom.key in player.
keyring:
     print(currentroom.key.keyusedesc)
```



```
West of House                                   Score: 0      Moves:  3

ZORK I: The Great Underground Empire
Copyright (c) 1981, 1982, 1983 Infocom, Inc. All rights reserved.
ZORK is a registered trademark of Infocom, Inc.
Revision 88 / Serial number 840726

West of House
You are standing in an open field west of a white house, with a boarded
front door.
There is a small mailbox here.

>open door
The door cannot be opened.

>e
The door is boarded and you can't remove the boards.

>unlock door
What do you want to unlock the door with?

>key
You can't see any key here!

>
```

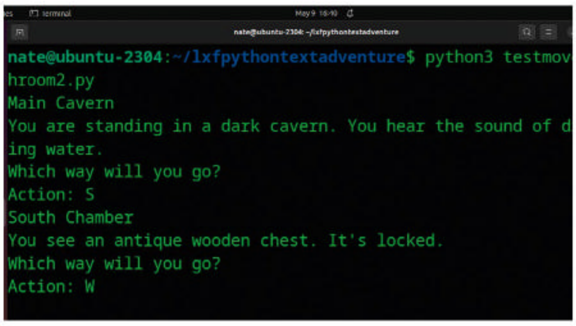
Famous text adventure games, such as Zork, have locked doors, making you either find the right key or another way in.

CREDIT: https://www.pcjs.org/software/pcx86/game/infocom/zork1/

```
    currentroom.description = currentroom.key.keyroomdesc
    currentroom.exits = currentroom.key.keyexits
    currentroom.key = None
checkkeys()
```

If the player has a key for a room, they'll see a message saying they've used the key. The room description and exits are also updated accordingly.

You can find a full example of this script at **https://tinyurl.com/lxypythontestmove4/**.



```
nate@ubuntu-2304:~/lxfpythontextadventure$ python3 testmov
hroom2.py
Main Cavern
You are standing in a dark cavern. You hear the sound of d
ing water.
Which way will you go?
Action: S
South Chamber
You see an antique wooden chest. It's locked.
Which way will you go?
Action: W
```

Although this script doesn't appear any different to players, under the hood it's using the 'room' class to load new areas.

```
        print("You cannot go that way.")

def get_player_command():
  return input('Action: ')

move()
```

As you can see, the game begins by placing the player in **room1** (Main Cavern). The newly tweaked `move` function displays the current room name and description before asking you where to go.

Once you choose a direction, `move` changes the player's location to another room, then reloads the `move` command to read out the name and description of that room. Try running the script and moving between rooms so you can see the code in action.

You can download an example of this script from **https://tinyurl.com/lxfpythontestmove2/**.

Where's the exit?

You may have noticed that even with our **room** and **player** objects nicely defined, you can still move from any room to any other. You can get around this by updating the **room** class to have clearly defined exits:

```
class room:
  def __init__(self, name, description, exits):
     self.name = name
     self.description = description
     self.exits = exits
```

This means when you define your **room** object, you also need to list the exits for them. You can do this using a list of values. Set them to **None** for now:

```
room1 = room(1, "Main Cavern", "You are standing in a dark cavern. You hear the sound of dripping water.",[None,None,None,None])
```

We do this because Python can't recognise objects until you declare them in the script. Once you've listed all the rooms, you can then add the exit rooms. There are four different list values to represent the four points of the compass (N, S, E and W), for example:

```
room1.exits = [room2, room3, room4, room5]
```

The **exits** attribute now shows that the player can go north, south, east and west into **room2**, **room3**, **room4** and **room5** respectively.

You can then update the `move` function to check for valid exits:

```
def move():
  print (player.currentroom.name)
  print (player.currentroom.description)
  print("Which way will you go?")
  while True:
    action_input = get_player_command()
    if action_input in ['n', 'N'] and player.currentroom.exits[0] != None:
        player.currentroom = player.currentroom.exits[0]
        move()
    elif action_input in ['s', 'S'] and player.currentroom.exits[1] != None:
        player.currentroom = player.currentroom.exits[1]
        move()
    elif action_input in ['e', 'E'] and player.currentroom.exits[2] != None:
        player.currentroom = player.currentroom.exits[2]
        move()
    elif action_input in ['w', 'W'] and player.currentroom.exits[3] != None:
        player.currentroom = player.currentroom.exits[3]
        move()
    else:
        print("You cannot go that way.")

def get_player_command():
  return input('Action: ')
```

The function has been modified so that when the player inputs a command, this must not only be a valid direction (N, S, E or W), but the corresponding **exits** value for the current room must be another room. If so, then the player is taken there.

You can download an example of this script with exits set up for all five rooms from **https://tinyurl.com/lxfpythontestmove3/**.

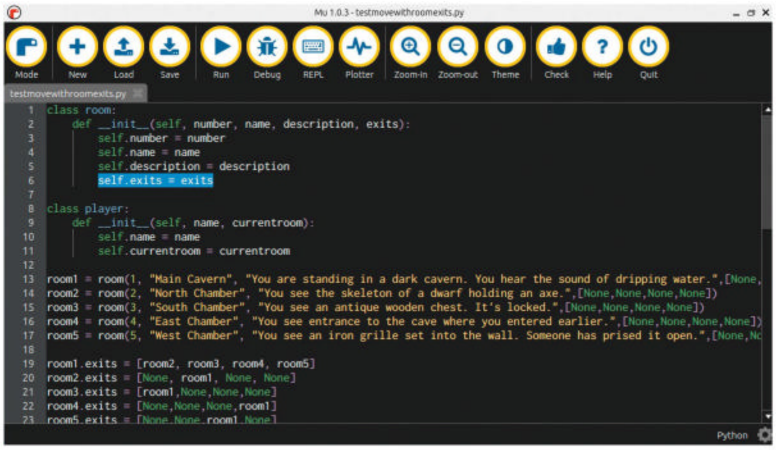
Dropping off the map

By now, believe it or not, you've learned almost all the essential code to create your very own Python text adventure. The remaining parts of this series will make good use of the code you've seen so far to allow your player to interact with items, fight the bad guys and even add some simple graphics.

If you run into any errors in your interpreter, make sure all your variables and commands are correctly spelled. Python also places great stock in indented code. If you're still having difficulties, feel free to download and run a copy of all the code so far from **https://tinyurl.com/lxypythontestmove4/**. LXF

QUICK TIP

For simplicity, we've placed all the code for the game in one simple script. A competent Python programmer would probably split a game into several files then call on them from the main script. Read more at: https://docs.python.org/3/tutorial/modules.html



```
class room:
    def __init__(self, number, name, description, exits):
        self.number = number
        self.name = name
        self.description = description
        self.exits = exits

class player:
    def __init__(self, name, currentroom):
        self.name = name
        self.currentroom = currentroom

room1 = room(1, "Main Cavern", "You are standing in a dark cavern. You hear the sound of dripping water.",[None,
room2 = room(2, "North Chamber", "You see the skeleton of a dwarf holding an axe.",[None,None,None,None])
room3 = room(3, "South Chamber", "You see an antique wooden chest. It's locked.",[None,None,None,None])
room4 = room(4, "East Chamber", "You see entrance to the cave where you entered earlier.",[None,None,None,None])
room5 = room(5, "West Chamber", "You see an iron grille set into the wall. Someone has prised it open.",[None,Nc

room1.exits = [room2, room3, room4, room5]
room2.exits = [None, room1, None, None]
room3.exits = [room1,None,None,None]
room4.exits = [None,None,None,room1]
room5.exits = [None,None,room1,None]
```

By adding the 'exits' attributes to the 'room' class, you can specify which rooms can be accessed from others.

RICH

Credit: https://github.com/Textualize/rich

Part Two! Subscribe on page 16 to access past issues!

Spice up your Python console applications

Matt Holder demonstrates how to use the Python Rich library to add formatting to your console applications.

OUR EXPERT

Matt Holder has been a fan of the open source methodology for over two decades and uses Linux and other tools where possible. In his spare time, he enjoys listening to music and reading.

In part two of our series on the Rich library, we are working through how to split our terminal application into sections, using layouts, outlining information with panels, rendering links and markdown, and using the helper log function to render logs with the time the message occurred.

Each of the code samples below can be run separately or in one file, and they are also available from GitHub *(see the Quick Tip)*. If you didn't catch last month's tutorial, the library can be installed using the following command from a terminal: `pip install rich`.

We use *Visual Studio Code* – it's an excellent way to develop, debug and run code; follow the instructions at **https://code.visualstudio.com/docs/setup/linux**. When creating files in *VSCode,* make sure that any files are saved with a .py extension. Files can be run by pressing Ctrl+F5. If developing using another text editor, once the files have been saved, they can be run by entering `python3 FILENAME.py` in the terminal.

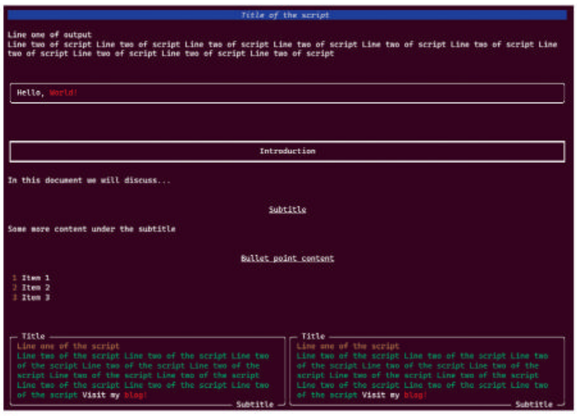
First, let's look at the logger function. This sample code uses a combination of the built-in Python logging functionality and functionality from the Rich library, which colour codes the output based on severity. Create a file called **logTest.py** and enter the following:

```
import logging
from rich.logging import RichHandler
FORMAT = "%(message)s"
logging.basicConfig(
  level="NOTSET", format=FORMAT, datefmt="[%X]",
handlers=[RichHandler()]
)
log = logging.getLogger("rich")
log.debug("Hello, World!")
log.info("Hello, World!")
log.warning("Hello, World!")
log.error("Hello, World!")
```

Here we first import the required libraries. On the next line we define the format of each log message, which is to simply write the message itself. On the next line, we configure the logging system to display the message in the defined format, display the date with it and use the Rich logging function to display the message. Before writing log messages, we need to create a **logger** object and then we reference it when writing log messages. Save the file and run it; you will see messages in green, blue, orange and red.

QUICK TIP

The code can be downloaded from https://github.com/mattmole/LXF303-304-Python-Rich



```
                                 Title of the script

Line one of output
Line two of script Line two of script Line two of script Line two of script Line two of script Line two of script Line
two of script Line two of script Line two of script Line two of script

 Hello, World!

                                     Introduction

In this document we will discuss...

                                       Subtitle

Some more content under the subtitle

                                  Bullet point content

 1 Item 1
 2 Item 2
 3 Item 3

 Title                                           Title
 Line one of the script                          Line one of the script
 Line two of the script Line two of the script Line two    Line two of the script Line two of the script Line two
 of the script Line two of the script Line two of the      of the script Line two of the script Line two of the
 script Line two of the script Line two of the script      script Line two of the script Line two of the script
 Line two of the script Line two of the script Line two    Line two of the script Line two of the script Line two
 of the script Visit my blog!                    Subtitle  of the script Visit my blog!                    Subtitle
```

Examples of our code. Note how text can be coloured, as we saw last month, and links can be clicked on with Ctrl+click.

Now let's move on to the justify options to change where text is rendered. Create a file called **justifyText.py** and enter the following code:

```
from rich.console import Console
console = Console(width=100)
style = "italic white on blue"
console.print("Title of the script", style=style,
justify="center")
console.print("")
console.print("Line one of output")
console.print("Line two of script "*10)
```

In this sample, we first import the necessary functions before defining a **console** object, which is where we print everything. We determine that the width should be 100 characters. Next we define a style, a preset built into the Rich library Then we call the **console.print()** function. This ensures that the **console** object is used to display any messages. When calling the **print** function, we are writing our message, then defining the style to be used. Finally, we call the **justify** argument to centre the text (notice the US spelling in the code). On the final three lines of output, we write a blank line, then another line, before repeating the final

» LIVE UPDATES

Everything we've investigated so far in this series of articles has been formatting static content. The Rich library also has functionality to allow live updates of the screen. This could be used to count the number of files in a folder, keep track of a copy operation, or maybe make sure that hard drives don't get filled up while downloading large files. Create a file called **liveTable.py** and add the following code:

```
import random
import time
from rich.table import Table
from rich.live import Live
def generate_table():
  “””Make a new table.”””
  table = Table()
  table.add_column(“ID”)
  table.add_column(“Value”)
  table.add_column(“Status”)
  for row in range(random.randint(2, 6)):
    value = random.random() * 100
    table.add_row(
      f”{row}”, f”{value:3.2f}”, “[red]ERROR” if value < 50 else “[green]SUCCESS”
    )
  return table
with Live(generate_table(), refresh_per_second=4) as live:
  for _ in range(40):
    time.sleep(0.4)
    live.update(generate_table())
```

In this sample, we import the necessary libraries then create a function, which draws a table and populates it with random values. Instead of random values, see the previous ideas for what could be coded. Next we use the **with** functionality to create an object called **live**, which is of type **Live**. We then use a **for** loop to update the table 40 times. Whenever the **live** object is updated, we call the **generate_table** function to generate the table with new values. Running the code shows this functionality.

message 10 times. Run the code and you will see output on the screen.

Our third example looks at the use of panels. These can be used to outline renderable items, such as tables, text and markdown. Create a file called **renderPanel.py** and enter the following code:

```
from rich import print
from rich.panel import Panel
import os
print(Panel(“Hello, [red]World!”))
textToDraw = “[yellow]Line one of the script[/yellow]” + os.linesep
textToDraw += “[green]Line two of the script [/green]”*10+os.linesep
textToDraw = textToDraw[0:-1]
textToDraw += “Visit my [red][link=https://www.linuxformat.co.uk]blog[/link]!”
panel = Panel(textToDraw)
panel.title = “Title”
panel.title_align = “left”
panel.subtitle = “Subtitle”
panel.subtitle_align = “right”
print(panel)
```

After importing the libraries we need, we use the **print** function to display the argument to the screen. This consists of the **Panel** function, which itself takes a message as the argument. Next we define a variable called **textToDraw**, which contains a number of strings, separated by your operating system's return character (note the use of **os.linesep**). On the third line of adding information to the **textToDraw** function, we return all but the final character, using the slice operator to remove the final space character. Next, we add another line, with a link to the *Linux Format* website. Then we create a **panel** object and pass the **textToDraw** variable as an argument. The next four lines set title, subtitle and alignment, before we print this panel to the screen.

Lastly, let's investigate how to read in markdown from a file and display this along with some other information in a layout. We'll split the screen into two rows and the bottom row will have two columns. Create a file called **printLayout.py** and add the following code:

```
from rich import print
from rich.console import Console
from rich.markdown import Markdown
from rich.layout import Layout
from rich.panel import Panel
markdownContent = open(“markdownForTutorial.md”).read()
md = Markdown(markdownContent)
console = Console(width=100)
layout = Layout()°
layout.minimum_size = 28
topRow = Layout(md, name=”upper”)
topRow.size = 20
bottomRow = Layout(name=”lower”)
bottomRow.size = 8
bottomLeft = Layout(panel, name=”left”)
bottomRight = Layout(panel, name=”right”)
layout.split_column(
  topRow,
  bottomRow
)
layout[“lower”].split_row(
  bottomLeft,
  bottomRight,
)
console.print(layout)
```

Save the file and run it. We first load in some sample markdown from a file and read the contents, then pass this variable to the **Markdown** function and call it **md**. Then we create a **console** object and set its width. Next we create a new layout and set the **minimum_size** attribute to 28 lines. We create two more **Layout** objects – the first contains the markdown object and the second is for the bottom row. We define two more **layout** objects, one for bottom-left and one for bottom-right. Then we take all the **layout** objects and assign them to the relevant locations within the first **layout** object. The **layout.split_column()** function splits the **layout** object into two rows and assigns the two row layouts to it. Next, we use the **split_row()** function to split the bottom row into two columns. Finally, we use **console.print()** to print the layout to the screen.

QUICK TIP

Further information can be found at https://rich.readthedocs.io/en/stable/

NEXT MONTH

LXF305 will be on sale Tuesday 25th July 2023

CREDIT: Getty Images

OWN YOUR OFFICE!

Reclaim your document liberty, retake your data, control your files and run your own office.

Running RISC-V

We get you up and running with the lowest-cost RISC-V solution on the planet. Run the next-gen ISA for sub-Pi prices!

Firewall in the house

Fire up your own personal network-wide firewall as we install the best open source deployment for home protection.

Computer-aided design

Build models for engineering, home architecture and 3D printing; we test the best open source CAD solutions.

Missile Command!

Any excuse to code a classic, we show you how to set up a Raspberry Pi dev station and code your own shooter.

Contents of future issues subject to change. Who are we kidding? We never go into the office!

LINUX FORMAT
The #1 open source mag

Future Publishing Limited,
Quay House, The Ambury, Bath, BA1 1UA
Email contact@linuxformat.com

EDITORIAL
Editor-in-chief Neil Mohr
Art editor Fraser McDermott
Production editor Katharine Davies
Group editor-in-chief Graham Barlow
Group art director Jo Gulliver
Editorial contributors Paul Alcorn, Mats Tage Axelsson, Mike Bedford, Denise Bertacchi, Neil Bothwick, Nate Drake, Darien Graham-Smith, Tam Hanna, Matthew Holder, Christopher Livingston, Nick Peers, Les Pounder, Michael Reed, Mayank Sharma, Shashank Sharma
Cover illustration Magictorch.com
Ubuntu is a trademark of Canonical Limited. We are not endorsed by or affiliated with Canonical Limited or the Ubuntu project. Raspberry Pi is a trademark of the Raspberry Pi Foundation. Tux credit: Larry Ewing (lewing@isc.tamu.edu).
Content production Adequate Media Limited

ADVERTISING
Commercial sales director Clare Dove
clare.dove@futurenet.com
Advertising director Lara Jaggon
lara.jaggon@futurenet.com
Account director Andrew Tilbury
andrew.tilbury@futurenet.com

INTERNATIONAL LICENSING
Head of print licensing Rachel Shaw
Linux Format is available for licensing and syndication. To find our more contact us at licensing@futurenet.com or view our content at www.futurecontenthub.com

NEW SUBSCRIPTIONS & BACK ISSUES
Web www.magazinesdirect.com
UK 0330 333 1113 **World** +44 (0) 330 333 1113

EXISTING SUBSCRIPTIONS
Web www.mymagazine.co.uk
UK 0330 333 4333 **World** +44 (0) 330 333 4333
Subscription delays: Disruption remains within UK and international delivery networks. Please allow up to seven days before contacting us about a late delivery to **help@magazinesdirect.com**

CIRCULATION
Newstrade & retail category director Ben Oakden

PRODUCTION AND DISTRIBUTION
Group head of production Mark Constance
Production manager Nola Cokely
Senior ad production manager Jo Crosby
Digital editions manager Jason Hudson

THE MANAGEMENT
Managing director technology group Paul Newman
Global head of design Rodney Dive
Commercial finance director Tania Brunning
Printed by Wyndeham Peterborough, Storey's Bar Road, Peterborough, Cambridgeshire, PE1 5YS
Distributed by Marketforce, 5 Churchill Place, Canary Wharf, London, E14 5HU
www.marketforce.co.uk For enquiries email: mfcommunications@futurenet.com

Disclaimer All contents © 2023 Future Publishing Limited or published under licence. All rights reserved. No part of this magazine may be used, stored, transmitted or reproduced in any way without the prior written permission of the publisher. Future Publishing Limited (company number 2008885) is registered in England and Wales. Registered office: Quay House, The Ambury, Bath BA1 1UA. All information contained in this publication is for information only and is, as far as we are aware, correct at the time of going to press. Future Publishing Limited cannot accept any responsibility for errors or inaccuracies in such information. You are advised to contact manufacturers and retailers directly with regard to the price of products/services referred to in this publication. Apps and websites mentioned in this publication are not under our control. We are not responsible for their contents or any other changes or updates to them. This magazine is fully independent and not affiliated in any way with the companies mentioned herein.

If you submit material to us, you warrant that you own the material and/or have the necessary rights/permissions to supply the material and you automatically grant Future Publishing Limited and its licensees a licence to publish your submission in whole or in part in any/all issues and/or editions of publications, in any format published worldwide and on associated websites, social media channels and associated products. Any material you submit is sent at your own risk and, although every care is taken, neither Future nor its employees, agents, subcontractors or licensees shall be liable for loss or damage. We assume all unsolicited material is for publication unless otherwise stated, and reserve the right to edit, amend and adapt all submissions. All contents in this magazine are used at your own risk. We accept no liability for any loss of data or damage to your systems, peripherals or software through the use of any guide. Note: Produced from a tent in Dorset!

We are committed to only using magazine paper derived from responsibly managed, certified forestry and chlorine-free manufacture. The paper in this magazine was sourced and produced from sustainable managed forests, conforming to strict environmental and socioeconomic standards.

Widely Recycled

Future is an award-winning international media group and leading digital business. We reach more than 57 million international consumers a month and create world-class content and advertising solutions for passionate consumers online, on tablet & smartphone and in print.

FUTURE
Connectors. Creators. Experience Makers.

Future plc is a public company quoted on the London Stock Exchange (symbol: FUTR)
www.futureplc.com

Chief Executive Officer **Jon Steinberg**
Non-Executive Chairman **Richard Huntingford**
Chief Financial and Strategy Officer **Penny Ladkin-Brand**

Tel +44 (0)1225 442 244

ipso. For press freedom with responsibility

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