



# RATED: PRIVACY DISTROS

Stay safe online with >> **1** Kodachi 7.1 >> **2** Tails 4.8  
>> **3** Septor 2020.03 >> **4** TENS 3.0.1 >> **5** Whonix 15

OF FREE EBOOKS!  
**£50**

# LINUX FORMAT

**PLUS!**  
**HOW TO...**

- EMULATE THE C64
- REPLACE YOUTUBE
- SOLVE LINUX PROBLEMS

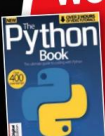
**4.3GB  
DOWNLOAD**  
EndeavourOS  
KaOS 2020

## BACK UP LIKE CLOCKWORK!

The ultimate guide to protecting your files and making effortless snapshots

**FREE!**  
Worth £50

**5 top ebooks**  
Over 700 pages



**57** pages of tutorials & features

Discover Godot, the open source gaming engine

Catalogue DVDs, books and more the easy way

Schedule tasks in the terminal like a pro!



Collabora

**Simon McVittie**

on life behind the DBus and keeping Steam running

### LIBRE OFFICE 7.0

Huge update! Get all the juicy new features

### THE GAME OF LIFE

How to code the classic cellular game of evolution

LXF September 2020

FUTURE



# PAPER POWER

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

Discover the story of paper

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

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







# LINUX FORMAT



## » WHO WE ARE

This issue we're trying to get readers to back up like clockwork, so what's your tale of woe about losing something vital in a system failure?



**Jonni Bidwell**

That feeling that you saw something in a tweet half an hour ago, but after scrolling through your and other people's timelines it's nowhere to be found. Both witty tweet and precious hours are lost, and all you've got to

show for it is some poor-to-average memes and a healthy dose of Twitter rancour. Also, one day I smashed two phones.



**Nick Peers**

Back in 1999 I managed to wipe the Master Boot Record on a friend's hard drive that contained hundreds of text files – all irreplaceable. Luckily, I found a data-recovery tool and spent most of that Christmas

holiday recovering each file to a floppy disk, one at a time.



**Les Pounder**

I had a 2TB drive die, no warning. There were no backups. I bought a new drive and spent a weekend recovering data from the old to the new drive. I had 30 minutes, then the drive failed and I had to start again. I managed to recover 90 per cent of the files.



**Calvin Robinson**

I once lost all my MP3s – we're talking a decade's worth of music. This was in the days before Spotify and Apple Music. All my soundtracks, albums, singles... everything was gone! It was like losing the

family photo albums in a house fire.



**Alexander Tolstoy**

Way back in the times of CD-RW drives I used to keep a backup of family photos on a disc, which I left under direct sunlight on my windowsill. When I tried to read it just a few minutes later, it was already useless!

## Backups are dead, long live backups!



No one needs to back up any more. Yes, after decades of tech "journalists" telling you to back up your systems, modern ways of working have made the backup obsolete. No, wait, don't go! We're not obsolete, just yet. It's true enough that many of us are working in "the cloud" and have work assets and media automatically backed up with some unknown level of redundancy.

Yet how can you do that yourself? How can you protect your system?

How can you make backups of everything run like clockwork, while making any data recovery just as slick? We're here to show you how. From traditional backups, to cloud backups and on to automatic snapshots, we'll cover a solution that will meet your needs. While it's good to back up, it's better to have a recovery plan, too!

We're packing the rest of the issue with all the great free and open source topics we know you love. Our *Roundup* is looking at the popular subject of online privacy, testing the latest versions of anonymising distros. We're sure some will disagree, so make sure you let us know what you use and why...

We're also pushing the release of *LibreOffice 7* that's now out. We're checking out the latest evolution and the new features, while in *News* we catch up with the new direction that The Document Foundation wants to take the *LibreOffice* ecosystem, to help raise revenue for *LibreOffice* developers. So give it a try and our packed tutorial section. Enjoy!

**FREE!**  
**Worth £50**  
5 top ebooks  
Over 700 pages



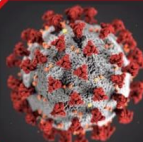
*Neil*

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



**Subscribe  
& save!**

On digital and print  
– see p22



Due to the unprecedented circumstances created by the Covid-19 outbreak, we've been forced to reduce the page count of *Linux Format* and drop the DVD. We wholeheartedly apologise for this – it's certainly something we do not want to do. The real hope is we can resume normal business for **LXF270**. Stay safe.



# Contents



## REVIEWS

### AMD Ryzen 3 3300X ..... 17

Is this the Goldilocks processor? Well, **Zak Storey** did catch it stealing porridge from Intel, so the moniker might stick...



### Crucial P2 500GB ..... 18

What's this? Low-cost SATA-breaking speed without the DRAM? **Sean Webster** is impressed, albeit with a few reservations.

### EndeavourOS 7.15.2020 ..... 19

Just as HMS Endeavour explored the unknown southern lands, so **Jonni Bidwell** uses EndeavourOS to explore Arch.

### Bunsenlabs Lithium ..... 20

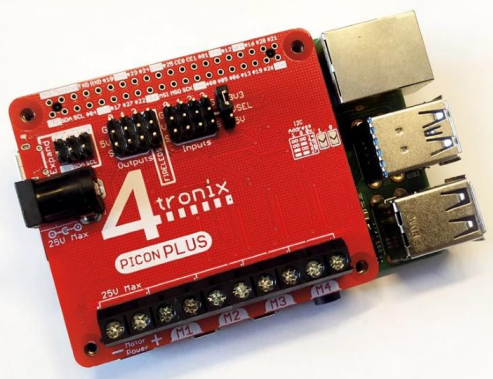
Forever grunge **Jonni Bidwell** is pleased to find a distribution that's almost as good as the Nirvana song. Almost...

### Tropico 6 ..... 21

Management are terrified of the new *Linux Format* workers' council, but it's just **Phil Savage** playing at communism.

### 4tronix Picon Plus ..... 49

**Les Pounder** needs to build a robot to help during social isolation. But can he build a robot butler to deliver more Pis to his table?



## BACK UP LIKE CLOCKWORK!



**Jonni Bidwell** has a plan – several plans, in fact – to keep his ever-growing repository of data ticking over nicely. The back-up fun begins on page 30.

## ROUNDUP



### Privacy distributions ..... 24

Only Batman can give you "clean slate" software, but if it's anonymity you're after, **Shashank Sharma** knows of some distributions that can help.

## INTERVIEW



### Pressure Vessel ..... 38

**Jonni Bidwell** gets on the video chat to talk video games and message buses with D-Bus guru **Simon McVittie**, while fondly reminiscing over maths group theory.



## PI USER

### Raspberry Pi news ..... 48

The Raspberry Pi takes on Amazon's all-conquering Web Services, we marvel at a Pi radio project that makes the world turn, and the results of the Astro Pi competitions are announced.

### 4tronix Picon Plus ..... 49

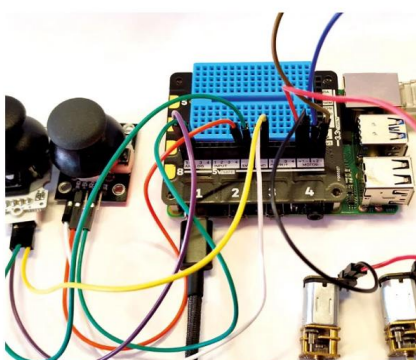
**Les Pounder** needs to build a robot to help during social isolation. But can he build a robot butler to deliver more Pis to his table?

### Basic touch and motor controls ..... 50

Get tinkering with **Les Pounder** as he demonstrates two projects using the Explorer HAT Pro board. It may be old, but is by no means obsolete!

### How to run Arch Linux on the Pi ..... 52

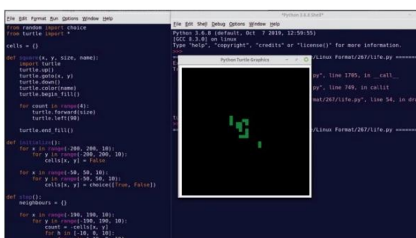
**Jonni Bidwell** installs Arch Linux in less than three days and discovers the best way to browse the web is from the console.



## CODING ACADEMY

### Recreate the Game of Life ..... 78

**Calvin Robinson** uses Python and Turtle graphics to recreate this unique game that tries to recreate life itself.



## REGULARS AT A GLANCE

### News ..... 6

LibreOffice's future development plans raises hackles, Tor Project could be denied funding, could encrypted data be a thing of the past, Linux Foundation helps out in pandemic, Manjaro shake-up and more!

### Your free ebooks ..... 11

### Answers ..... 12

Encrypt a second disc drive, solving random crashes, installing a distro from a USB stick, and running terminal commands remotely.

### Mailserver ..... 14

Readers ask us for advice on Ubuntu-compatible laptops, what open source programs can replace Adobe's Creative Cloud suite, and Bluetooth battery usage.

### Subscriptions ..... 22

Take out a subscription to *Linux Format* and receive a stylish USB charger.

### Hot Picks ..... 63

**Alexander Tolstoy** hasn't got time to head over to East Russian to organise anti-Putin protests, he far too busy organising only the best behaved open source such as: *Symphonium*, *Cadmus*, *E4head*, *Gromit-MPX*, *Fsmon*, *Qt-SESAM*, *Video-trimmer*, *Pillar Valley*, *CaveExpress*, *Flatpak* and *Viper Browser*.

### Back issues ..... 69

They're selling out fast!

### Next month ..... 82

But you've hardly read this month's issue!



### Your free ebooks ..... 11

Claim your four free ebooks and discover how you can get your digital download and what the heck to do with an ISO file!

## TUTORIALS

### TERMINAL: Scheduling tasks ..... 54

Want to make time for hobby projects such as uncovering the name of Doctor Who?

**Shashank Sharma** suggests automating recurring tasks with *Cron*.

### DATABASES: Symphytum ..... 56

**Nick Peers** reveals how to create your own simple databases in minutes using *Symphytum* – no coding or server experience required!

### MEDIAGOBLIN: Streaming video ..... 60

**Mike McCallister** helps you collect and share your sporting exploits with the free video service *MediaGoblin* and a suitable amount of terabyte storage.

### EMULATION: Commodore 64 ..... 70

**Les Pounder** reveals how to relive the 1980s with a Commodore 64 and BASIC via emulation, without the rubbish hair, but with all the demos.

### GODOT: Create a video game ..... 74

Following on from last issue's introduction to the development engine Godot, **Calvin Robinson** shows you how to develop your own game.

## IN-DEPTH

### LibreOffice 7 ..... 44

**Neil Mohr** discovers what's new in this latest release of the office suite that's one of the crown jewels of the open source world.



# Newsdesk

**THIS ISSUE:** Concerns over LibreOffice » Changes in Open Technology Fund » Not-so secret data » Linux tackles Covid-19

## OPEN SOURCE OFFICE

# LibreOffice looks to protect long-term viability

Community hackles are raised as The Document Foundation markets a LibreOffice ecosystem that encourages greater corporate engagement.

**T**here was a brief internet drama back in July when a release candidate for *LibreOffice 7.0* appeared with the tag of “Personal Edition”. Some people assumed this version lacked features that some other, yet-to-be-released edition may sport. This then led to the conclusion that The Document Foundation – the non-profit body that oversees *LibreOffice*’s development – was moving the suite to some other type of licencing model.

This was not the case. Initially, TDF’s response was to propose using a “Community Edition” tag that better expresses the intent. Now the naming plans have been put on hold until after *LibreOffice 7.0* is released.

The worry is understandable; many open source projects have changed direction to enable better commercial support of the core product. One model is called Open Core and is an example of a dual-licence system, where a open source “core” of features is released, while a proprietary paid-for set is commercialised. Services such as *Kafka*, *Cassandra*, *GitLab* and *Redis* are all successful examples of this. There was a genuine concern that this was the plan for *LibreOffice*, too.

We contacted The Document Foundation to ask what was going on and Mike Saunders, marketing assistant at TDF (yes, of *ex-Linux Format/Linux Voice* fame) explained this was part

of a new marketing strategy to try and support third-party *LibreOffice* developers such as Collabora and CIB, which provide support and products based on *LibreOffice* and *LibreOffice Online*.

The issue is that while *LibreOffice* has a large (200 million) user base, there’s almost no paid-for commercial enterprise use. That’s not to say enterprises and businesses of all sizes don’t use *LibreOffice* – they certainly do as a one-off install – but then don’t pay for support. They still post bug reports, but offer no developer support.

Tagging a version downloaded for individual use as Personal or Community Edition was the first step in a process to drive businesses and enterprises towards using third-party supported service options. It’s these companies that deliver much of *LibreOffice*’s development, rather than just leeching off any freely available download, mass-installing and abandoning it.

It’s a tricky balancing act. On the one hand, TDF is legally set up as a non-profit body. On the other, it wants to help promote developers that can offer commercial services that go on to help the entire *LibreOffice* ecosystem... all the while reassuring the community this is all in its best interests. You’d hope TDF’s benevolent and positive governance track record will help ease concerns as it takes steps in this new direction.

For more about the problems faced by TDF and others, see Michael Meeks’s enlightening post at <https://lwn.net/Articles/825602>.



**LibreOffice**  
The Document Foundation

A key part of any open source operating system.



**Collabora  
Productivity**

Collabora is  
the largest  
contributor to  
LibreOffice.



OPEN INTERNET

# Tor Project under threat

The Trump Administration is holding back Congress-approved funding.

**Y**ou may never have heard of the US Open Technology Fund (OTF), but it's a US digital rights non-profit body that's funded by the US government. The OTF helps fund a wide range of projects that enable people in repressed regimes – and the UK – to safely access the whole of the internet, including pro-US news outlets. Such projects include Signal, Tails, Tor, Wireguard, DNS Privacy, No Script and Qubes (more projects are listed at [www.opentech.fund/results/supported-projects](http://www.opentech.fund/results/supported-projects)).

So it seems bizarre that this body is under close scrutiny from the Trump administration. The previous CEO, Libby Liu, and the entire board of the OTF was fired by Trump appointee Michael Pack out of the blue with no justification, while a \$9 million Congress-approved payment has been withheld by the White House.

Letters sent to involved parties indicate current OTF funding should now be redirected to

just four projects – two of which, Ultrasurf and Freagate, are pro-Trump and closed-source. This significantly narrows the scope of the project and appears mostly to be redirecting funds to private, pro-administration businesses.

See what the Tor Project has to say on this here: <https://blog.torproject.org/save-open-technology-fund>, while the Electronic Frontier Foundation's take on this development can be read at [www.eff.org/otfs-work-is-vital](http://www.eff.org/otfs-work-is-vital).



Paying your mates always works out well in the long run...

ENCRYPTION SHENANIGANS

# Encrypted Data Act

The US Congress is attempting to outlaw privacy – again.

**C**onspiracy theorists often say governments love spying on their citizens. Yet the US Congress is living up to that label, as it tries to pass a law that forces private business to place government-accessible backdoors in any encryption systems. The latest bill called the Lawful Access to Encrypted Data Act follows on from the recent EARN IT bill that had similar goals.

The bill would instruct companies to comply with decryption orders, while online services would need to have central access to all encrypted data, possibly with a general decryption key. When being presented with a court order, they would have to decrypt the data of this particular user, but leave the data of all other users untouched.

Data breaches are already at an all-time high, and that's without a central insecurity being in play, such as a industry-wide backdoor key. US citizens who think this is a bad idea should contact their Congressperson. Read more about the EARN IT bill at <https://act.eff.org/action/stop-the-earn-it-bill-before-it-breaks-encryption>.

Breaking free speech and breaking encryption keys. Naughty eagle!



CREDIT: EFF

OPINION

# RING THE PLUMBER!



**George Kiagiadakis**  
Senior software engineer,  
Collabora

**“**A key improvement that *PipeWire* brings over *PulseAudio* is that the policy is separate from the media handling. The policy is the code that decides which audio device to use, which software has access to them and how it's all connected. In *PipeWire*, the policy engine is called a session manager. *WirePlumber* is the modular and extensible session manager that brings configurability and an abstraction layer to make it easier to create these policies.

Released late June, *WirePlumber* 0.3.0 brings support for desktop use cases. It introduces the use of session, endpoint and endpoint-stream abstractions to orchestrate the *PipeWire* media graph. My blog post on [collabora.com](http://collabora.com) is a good starting point to read about endpoints.

Also noteworthy in this release is the ability to arbitrate the capture of audio devices between *PipeWire* and *JACK*. If a *JACK* server is started while *PipeWire* is running or is found already running when *PipeWire* is started, *WirePlumber* releases the audio device that *JACK* wants to control and sets up the *PipeWire* *JACK* source and sink nodes so that audio in *PipeWire* is routed through *JACK* to access that audio device.



## OPINION

THAT'S  
NO WOLF

**Matt Yonkovit**  
Chief experience officer  
at Percona

“You may think of MongoDB as a wolf hiding in an open source sheepskin jacket, but there’s no denying the NoSQL database’s popularity. So when a new beta appears it can’t be ignored. If you’re a DBA you’ll love it: this is the maintenance release you’ve been waiting for since v3.4.

One prime example of the available fixes is the ability to perform resumable initial syncs. Before MongoDB 4.4, replacing a server that was chugging under a high load could mean potential failures. This was usually because the primary would take too long to serve the data to the new node while it was initial syncing.

Another helpful fix involves risky maintenance operations with Oplog during increased write loads. MongoDB uses Oplog to record all operations that modify data on your primary DB. Secondary members copy that Oplog and apply the changes. This means that it’s critical the Oplog is the right size.

While you’ve had the option since v3.6 to manually change the Oplog size without a restart in the admin DB, being woken up for such an event could feel... stressful. Fortunately, 4.4 adds an automatic minimum Oplog window config setting, so you can stay fast asleep. ”

## OPEN SOURCE APPS

Linux Foundation to track  
and trace Covid-19

Ireland is doing what the UK government couldn't.

Ireland's Health Service Executive (HSE) has announced that it's donating the code for its Covid tracker app called *COVID Green* as open source to The Linux Foundation. The HSE said, "This will enable jurisdictions worldwide to quickly build and

deploy their own contact tracing apps using a wildly successful proven base." The Ireland-developed tracing app had exceeded all expectations, with a million installs within the first 36 hours.

To drive development The Linux Foundation Public Health initiative has been created. In a statement its general manager Dan Kohn said that the main goal, "is building a global community of leading technology and consulting companies, public health authorities, epidemiologists, and other public health specialists, privacy and security experts, and individual developers."

Learn more about the contact tracing app at <https://bit.ly/lxf267covid>.

The up and running  
Irish Covid-19 track  
and trace app.



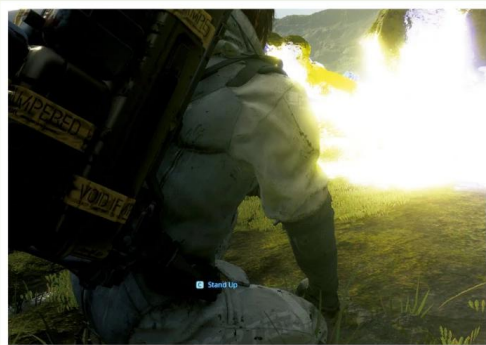
## GAMING

## Proton 5.0

Even more awesome.

It's a regular theme that we cover the latest milestone in Steam Gaming compatibility via the Valve developed Proton/Wine technology. While it might not be the most worthy of projects, the latest in-development *Proton 5* has been shown to run the DirectX 12 exclusive *Death Stranding* on AMD hardware, although the game suffers with lock-ups and visual issues, this is just an early beta!

CREDIT: Kojima Productions



*Death Stranding*: technically impressive, slow gameplay.

## DISTRO DRAMA

## Manjaro changes

Out with financial oversight.

The Treasurer of the Manjaro project has stepped down over a disagreement with Philip Müller, the project's founder. In a forum post that was initially removed, but later reinstated Jonathon Fernyhough explained that a €2,000 laptop purchase (plus other expenses) made by Müller on behalf of Manjaro dev Helmut Stult breached the existing Manjaro policy. A new expense policy is now in place and Manjaro goes rolling on. See the post here: <https://bit.ly/lxf267man>.



Manjaro goes  
from strength  
to strength



# Distro watch

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

## GECKOLINUX 999.200729.0

Lovers of reptile-based distros will love the latest spin of GeckoLinux, a rolling-release distro spun off openSUSE, Tumbleweed and Packman repositories. It's focused on giving a slick user experience out of the box, and includes restricted media codecs, hardware drivers and firmware. This latest release offers a fresh set of install ISO images for new users to get started with. More details here: <https://github.com/geckolinux/geckolinux-project/releases/tag/999.200729>.



## OPNSENSE 20.7 "LEGENDARY LION"

Try slipping a hardened firewall into your next dinner party chat and you might get to talk about OPNsense. This BSD-based distro is a dedicated firewall that's been around for over five years and offers an innovative modular design, simple firmware updates, multi-language support and quick integration of up-stream updates. Major updates for this release include DHCPv6, multi-WAN, a netstat tree view and live log filtering. See <https://opnsense.org/opnsense-20-7>.



## REMNUX 7

After 10 years of development the malware analysis tool REMnux offers a major update to version 7. Almost every part of this release is new and updated. All the tools have either been updated or retired with many new ones added. Ease of use has also been updated with all-new documentation while *SaltStack* is now used to install and configure the distro. Learn what's new at <https://zeltser.com/remnux-v7-release>.



## UNIVENTION CORPORATE SERVER 4.4-5

The Debian-based, enterprise-class server distro is now on version 4.4-5. This latest release much improves its single sign-on functionality via LDAP directory service improvements, as part of its path to a new 5.0 release for the end of 2020. This also included porting existing modules to Python 3 while there's a preview of the UCS 5 portal as part of the App Center. Visit [www.univention.com/blog-en/2020/07/release-ucs-4-4-5-2-brings-improvements](http://www.univention.com/blog-en/2020/07/release-ucs-4-4-5-2-brings-improvements) for more information.



## OPINION

## WANTS AND NEEDS



**Keith Edmunds**

MD at Tiger Computing Ltd, which provides support for businesses using Linux.

“We've just had an email from a client who wants a new, high-performance storage array to be shared between two nodes of a Linux cluster.

As always, there are things that they want, such as a given storage capacity, and there are things that they need. That latter list includes data integrity in the case of failover between the two nodes, data backups, resilience to hardware failure and more.

They've also specified an SSD caching layer. Is that a “want” or a “need”? I suspect neither. It would significantly increase the cost without appreciably affecting performance – a combination that's unlikely to appeal.

It's difficult when your users/clients/boss tells you what they want and how they want you to achieve it. Do you risk alienating them by telling them, essentially, that they're wrong?

I'd argue yes, albeit with diplomacy. It's our job, as Linux professionals, to establish and communicate the most cost-effective and efficient way of implementing the end result.

In the longer term, there's more value and trust in using our expertise and experience to suggest a better way when one exists.”



## OPINION

LINUX IS  
THE TOPS

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

Linux turned 29 shortly after we went to press. The world has changed considerably in the intervening years. Back in 1991 it may have been hard to imagine that the world's top 500 supercomputers would all be Linux based by 2020 (a record captured several years ago). And it wasn't all that long ago that the idea of running serious commercial workloads on Linux was shunned by self-respecting sysadmins and corporate execs alike.

Readers are perhaps familiar with the humble beginnings of Linux. If not, I encourage you to consult the Wikipedia page for a handy link to the original Usenet discussion that kicked it all off. At the time, microkernels were all the rage in academia, with Andrew Tanenbaum's Minix being the up-and-comer. By contrast, Linux followed a traditional (monolithic) single kernel image design that had fallen out of favour in academia.

Linus Torvalds' first message to **comp.os.minix** described his work as "just a hobby, won't be big and professional like gnu". In retrospect that sounds quaint today. A few folks on Twitter and elsewhere have been collecting stories from those who have been using Linux for a long time. Now would seem to be a good moment as there is a major milestone birthday coming up next year.

# Kernel Watch

**Jon Masters** summarises the latest happenings in the Linux kernel, so you don't have to.

Linus Torvalds announced the release of Linux 5.8, which has among its many headline features support for Branch Target Identification (BTI) and Shadow Call Stacks on Arm64. This will improve the security of many platforms, but especially future Android devices, including the prevention of control flow hijacking attacks using buffer overruns on the kernel stack. Linux 5.8 also includes an optimisation that should improve behaviour in memory "thrashing" situations, particularly when using contemporary in-memory swap devices.

While Linus had considered a 5.8-rc8, in the end he decided things were stable enough for the final release on schedule. And with the release of 5.8 comes the opening of the merge window (when disruptive changes are allowed) for what will become Linux 5.9. More on those upcoming features in next month's edition.

## The "Boothole" vulnerability

Several recent CVEs (aka security vulnerabilities) have been found in the GRUB2 bootloader commonly used to boot Linux systems. One bug in particular undermines the safety of the "Secure Boot" mechanism that's often used to lock down systems. A Secure Boot-enabled system typically can't run any random kernel, but only those signed by OS vendors, using a signed shim to verify GRUB which then verifies the OS has not been tampered with prior to booting the kernel.

The point of Secure Boot is to prevent a compromised laptop, server or other system from running an unsanctioned OS without being detected. In theory, an attacker could gain root on such a machine, but they would then be unable to replace the kernel or perform similar malicious activity without detection. A Secure Boot system would detect the modification and fail to boot with a suitable error message. The whole thing can be useful, but it has pitfalls.

First, there are many who hate the notion of Secure Boot on principle because it takes away some level of control. Second, Secure Boot is only as secure as the weakest link. If a vulnerability can be found in the signed pieces that are loaded, this can be exploited to undermine the whole thing. This is the case with "boothole", in which a one-liner mishandling of a "fatal" error in GRUB2 results in it continuing execution where it should have terminated.

A privileged user is able to create a GRUB2 configuration file that will cause this error to happen, not be handled correctly, and then permit a malicious payload to be run. Because Secure Boot relies on keys installed in shipping firmware on many millions of devices, the fix is rather painful. The entire Linux community had to "rekey" their signed shims, and the old keys must be added to a Revocation List handled by platform firmware known as dbx. The last step will prevent older OS media (such as installers) from running, so it has potential to be disruptive to those deploying systems at large scale. **LXF**

## » ONGOING DEVELOPMENT

Steven Price (Arm) posted a patch enabling MTE (memory tagging extension) for KVM guests. This means Linux will be able to support running virtual machines in which the kernel (and guest userspace) tags memory addresses in such a manner as to prevent malicious code from performing the same access. The end goal is to better thwart attackers who seek to exploit software bugs.

Sumeet Pawnikar (Intel) posted a patch titled Add Power Limit4 support, that exposes userspace controls over the PL4 (Power Limit4) on Intel mobile platforms. This is also known as the SoC package level maximum power, and modifying it (where safe to do so)

would seem to enable a user to increase maximum sustained performance through the commands in *powercap-utils*.

Christoph Hellwig recently became annoyed enough at the ongoing proprietary driver shenanigans being pulled by a particular large graphics card company, that he proposed a patch reworking the kernel's module tainting mechanism.

Once this patch is accepted into Linux, the kernel will refuse to load modules that in turn rely upon other modules that specifically disallow such uses. This effectively tries to prevent upstreaming "shims", which wrap binary drivers.



# Your free downloads

Distros, apps, games, books, miscellany and more...

## Get your free ebooks!

**T**o replace our regular covermounted DVD we're offering five top digital bookazines and a 4.3GB DVD ISO image for you to download.

Due to the Covid-19 pandemic we're unable to guarantee delivery of our usual *Linux Format* DVD (they're trucked to LXF Towers from Poland). So to avoid missing vital printer slots we're reluctantly dropping the physical disc until service can return to normal. As way of replacement we're still offering the DVD image

as a free digital download, but also throwing in £50 worth of top digital bookazines. With this issue we've got five of them for you to enjoy. To claim your digital bookazines just head to the following webpage and register: [https://bit.ly/lxf267\\_ebooks](https://bit.ly/lxf267_ebooks)<sup>1</sup>.

If you don't receive a response please try again and opt in to Future-only marketing – our GDPR filters are blocking some readers. Or get in contact direct on [linuxformat@futurenet.com](mailto:linuxformat@futurenet.com).

1. [https://www.surveymonkey.co.uk/r/Linux\\_August\\_ebooks](https://www.surveymonkey.co.uk/r/Linux_August_ebooks)



**FREE!**  
**Worth £50**  
**5 top ebooks**  
**Over 700 pages**

## Your digital download

**T**he usual *Linux Format* DVD has gone digital while the ongoing Covid-19 pandemic interrupts our normal service.

We've popped together the usual DVD in digital form and it's available as always to everyone in torrent form. Visit [www.linuxformat.com/archives](http://www.linuxformat.com/archives), download the torrent file and load

it into your favourite Torrent client, and then let the ISO download to your PC.

Once you have the ISO file you can use it to boot a virtual machine using, say, *VirtualBox*, use <http://Etcher.io> to write it to a suitably sized USB stick or – if you still have one – write the ISO directly to a blank DVD disc. More details at [www.linuxformat.com/dvdsupport](http://www.linuxformat.com/dvdsupport). **LXF**

**DOWNLOAD YOUR DVD**  
Get code and DVD images at:  
[www.linuxformat.com/archives](http://www.linuxformat.com/archives)

### EndeavourOS 15



**64-bit**

This is a rolling release distro based on Arch Linux, which provides a smooth online or offline install. It uses the Xfce desktop, so it's lightweight and fast.

**Minimum specification**  
64-bit CPU, 1GB RAM,  
12GB HDD, VGA display  
**Username:** N/A

**Recommended specification**  
Dual-core 64-bit CPU, 2GB  
RAM, 20GB HDD, HD display  
**Password:** N/A

### KaOS 2020.07



**64-bit**

An Arch-inspired rolling distro that's based on the KDE desktop with Qt toolkit. It uses its own packages and repros for a polished finish.

**Minimum specification**  
64-bit CPU, 1GB RAM,  
8GB HDD, SVGA display  
**Username:** N/A

**Recommended specification**  
Dual-core 64-bit CPU,  
2GB RAM, 25GB HDD  
**Password:** N/A

### » IMPORTANT NOTICE!

For basic help on running the disc image, please visit our support site at [www.linuxformat.com/dvdsupport](http://www.linuxformat.com/dvdsupport). Unfortunately, we're unable to offer advice on using the applications, your hardware or the operating system itself.

# Answers

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



**Neil Bothwick**  
is a man who's in the know (*taps forehead knowingly*).

**Q Second disk encryption**  
I have two SSDs in my machine running LXLE (Ubuntu 18.04). One is a boot/home drive that was encrypted at the time of install. I would now like to encrypt the second drive, which is used for data storage and VMs. I'm still a relative beginner and was wondering what's the most user-friendly way to encrypt the second drive and access it easily to run my VMs? There seems to be many options, for example *LUKS*, *GnuPG* and *VeraCrypt*.  
*Sandeep Bappoo*

**A** There are indeed many options, and we don't know which has been used to encrypt your main drive. When encrypting a whole disk or partition, *LUKS* is the usual choice. It's not suitable for a total novice to set up, but if you're happy to use the command line it's fairly straightforward. Encrypting individual directories is often done with *EcryptFS*. This is what Ubuntu uses if you choose an encrypted home directory when installing. *VeraCrypt* is different in that it runs in user space, has a GUI for set up and is cross-platform, so you can read the same encrypted data from Linux and Windows.

If you're already using *LUKS* for your first disk, you'll have a file in `/dev/mapper` that corresponds to your partition's entry in the output from `mount` or `df`. In that case you may as well use *LUKS* for the second disk, too. Setting up a partition with *LUKS* needs one command to encrypt it, so if the partition is `/dev/sdb1`, you'd use

```
$ sudo cryptsetup luksFormat /dev/sdb1
```

Now open it with

```
$ sudo cryptsetup luksOpen /dev/sdb1 mystuff
```

This creates a device at `/dev/mapper/mystuff`, which you would format and mount in the usual way.

```
$ sudo mkfs.ext4 /dev/mapper/mystuff -L mystuff
```

This is all a bit manual, so let's make life easier. First we'll add another passphrase to the encrypted device, but this time we'll use a file instead. The first command

creates a file of pseudo-random data as the passphrase, the second adds it to the encrypted container:

```
$ dd if=/dev/urandom of=mykey bs=512 count=1
$ sudo cryptsetup luksAddKey /dev/sdb1 mykey
```

Now you can unlock the container with either the passphrase or the key file. Store the key file somewhere safe, in your existing encrypted area, so it can't be read without unlocking the original encryption. Next, create a short script to open and mount it in one go. A script is just a list of commands in a file, so use your preferred text editor for this:

```
#!/bin/sh
sudo cryptsetup luksOpen /dev/sdb1 mystuff --key-file /path/to/mykey
sudo mount LABEL=mystuff /path/to/my/vms
```

Now you see why we used the `-L` option with `mkfs` to add a volume label. It saves having to know the exact path of the device. All that's left to do is set the executable property of the script, you can do this in your file manager and run it to make sure it works as expected. You can either add a desktop shortcut for run this on demand or add it to your desktop's autostart list to unlock and mount the disk when you log in.

This may seem a little complex, but most of the work is done only once and you

have a secure and efficient setup. On the other hand, if you want to do everything with a GUI, either use *VeraCrypt* or get someone else to set up *LUKS* for you.

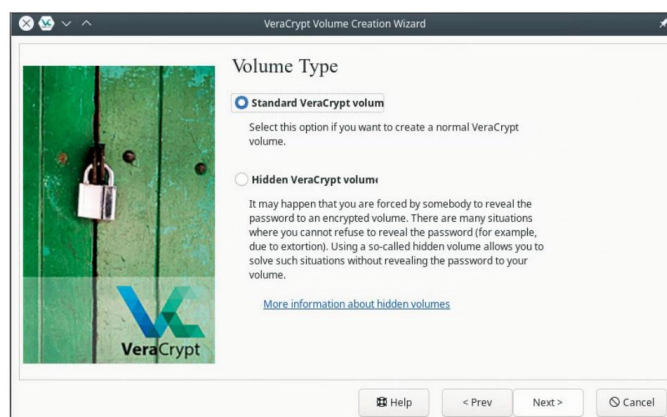
**Q Random(ish) crashes**  
I have an old Dell Studio 1555 dating from 2009, which has had several distros installed on it in turn. Recently though it's been crashing to a blank screen of mixed colours, sometimes randomly but often following a right-mouse click. Installing various distros from your cover DVDs has not cured it, so I'm thinking it may be a hardware failure rather than a software bug.

I was able to install Devuan as a live install from your DVD and used that to run `hardinfo`, as well as set *Bleachbit* going to wipe the hard drive in case it has to go to recycling. It may be time for it to go to recycling, although it was a nice machine with a fine screen. I don't want to waste time on it if it's failing – it has had plenty of use, most recently in the garden shed as a photo viewer and music player. Perhaps the damp got to it? What do you think?

*Matthew Hunt*

**A** Ten years is a good age for a computer, but I have a system here that's older than that and still running reliably, if somewhat slowly. Since the

crash isn't always immediately after a right-mouse click, I would think that isn't the cause, although the graphics update of popping up a menu may be involved. If we consider these crashes to be more random, there are four usual suspects that account for the majority of cases: poor power, heat, bad memory and



Consider VeraCrypt for disk encryption if you want ease of use and cross-platform access more than outright performance.

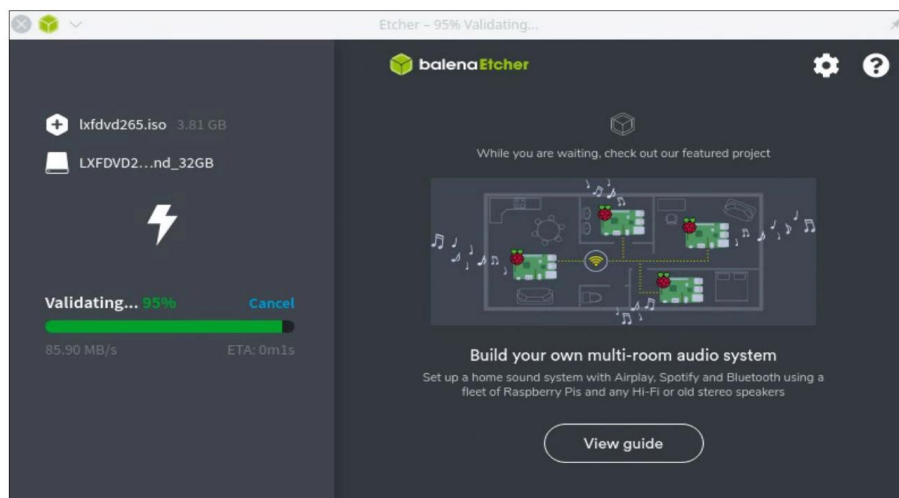


broken hardware. If we can track it down to one of the first three, you may still have life in your computer. I doubt a damp environment is the cause, especially if it's back indoors and has had time to dry out.

Because it's a laptop, poor power is unlikely to be the cause, although you could try running with no battery, or a replacement if you have one, to see if it helps. Are you comfortable opening up your computer? If so, you'll probably find 10 years of accumulated "dust bunnies" and general crud around your heat sink, fans and ventilation openings. Cleaning that out will reduce the temperature, and increase the life, of your computer. If you don't feel confident about opening it up, take a compressed air spray to the vents to blow out what you can. Be careful you don't breath in what you blow out (fortunately we all have face masks available these days).

Faulty memory is easily checked for by running *memtest86*, which you should find on our covers discs. Boot from the disk and let *memtest86* run for at least two or three passes (overnight is best). It's possible that the damp environment corroded the memory connectors, so if it does show errors, try removing and reseating the memory modules. Replacing memory chips is usually fairly easy – you're sure to find a demonstration on YouTube, but you may find the cost too much for an old computer. If *memtest86* finds no errors and the previous suggestions gave no joy, I'm afraid your hardware may be on its last legs.

**Q USB for LXF?**  
Is it possible to copy one of your Linux systems DVD to a USB flash drive



Etcher isn't only a friendly and safe way of copying images to flash drives, it'll also verify the data after writing.

**and boot a laptop computer with the flash drive, with the intention of installing the system into the laptop computer?**

**Terry**

**A** The short answer to your question is "yes". The *Linux Format* DVDs use what's known as a hybrid ISO format that enables them to be booted from a USB stick as well as an optical disc. This is also true of the majority of the ISO images on each DVD. So if you want to install one of the distros on the DVD from a USB stick, you only need to copy that distro.

The traditional way of doing this has been with the *dd* program, which unfortunately has no way of going back if you make a mistake with the options and wipe the wrong drive. These days we recommend the more friendly alternative, *Etcher* (<http://etcher.io>). This is supplied

as a zip file containing a single program file. Make that executable and run it and you'll have a graphical way of writing to a USB stick, and it will only write to removable devices so there's far less risk of overwriting the wrong information.

If you want to put the whole *Linux Format* DVD onto a stick, you'll first have to convert the disc back to an ISO with

```
$ dd if=/dev/sr0 of=lxfvdvd.iso bs=4M
```

You're copying to a file here, so the risks associated with using *dd* don't apply. */dev/sr0* is usually the first optical drive, but if your system uses */dev/dvd* or */dev/cdrom*, adjust the command accordingly. Now you can copy that ISO to your USB stick with *Etcher*. **LXF**

## GET HELP NOW!

We'd love to try and answer any questions you send to [lxf.answers@futurenet.com](mailto:lxf.answers@futurenet.com), no matter what the level. We've all been stuck before, so don't be shy. However, we're only human and so it's important that you include as much information as you can. If something works on one Linux distribution but not another, then tell us. If you get an error message, please tell us the exact message and precisely what you did to invoke it in the first place.

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
```

## » A QUICK REFERENCE TO FORWARDING

Running terminal commands on a different computer is easy, just SSH to the other computer and it's like you were there. You can also do the same with full desktops with VNC or RDP, but isn't that overkill for just one program?

The X graphical interface is a client/server system, so it is possible to run a program on one computer and open its window on another's desktop. This can be done using the SSH connection that you have already set up. You need this facility enabled on the remote computer. In */etc/ssh/sshd\_config*, ensure that **X11Forwarding** is set to **yes**. Restart *sshd* if you changed this setting, then on the local computer

connect over SSH as usual, but add the **-Y** option:

```
$ ssh -Y user@remotecomputer
```

Now try running a GUI program from the command line and it'll open its window on the local computer, but the program will be running on the remote system. Depending on your SSH configuration, you may need to use **-X** instead of **-Y**, but try **-Y** first because it's faster. On the subject of speed, if you're connecting over the Internet, adding **-C** to compress the traffic may speed things up, although *TightVNC* or *NX* may be faster over a slow Internet connection because they're optimised for this and X isn't.



# Mailserver

Write to us at Linux Format, Future Publishing, Quay House,  
The Ambury, Bath BA1 1UA or **lxf.letters@futurenet.com**.

## Better laptops

I'm looking for a new laptop to run Ubuntu on. I have a Dell Inspiron N5110, which works most of the time. If I hibernate it, the screen can go blocky, but a reboot fixes this. Could you curate a list of reader's laptops and what distros/versions they're running and any problems they have? Turns out PC World aren't too happy if you try to reboot their demo machines with a USB stick...

**Geoff Charlton**

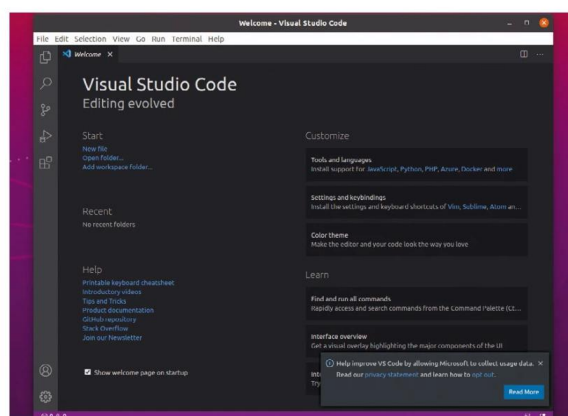
Neil says...

Your hibernation issue sounds like this bug report (<https://bit.ly/lxf267bug>). Low-power, sleep and hibernate features have always been a bit of a poor show with Linux, as often these are implemented between hardware manufacturers and Microsoft, meaning they're often plugged into Linux later on.

That's an interesting idea about a reader database but I wonder how useful it would be? There's already such things online and the problem is that they end up being woefully out of date. Furthermore, if most laptop models cannot be bought then how is it actually helping anyone? As in these will be old models I'd guess.

The best option is to file complete bug reports when issues arise and choose laptops that support Linux in the first place. Thankfully, this is getting easier!

Ensure your laptops works with Linux by buying a Linux laptop.



You don't have to choose the default Microsoft build of Code – there's a telemetry free version available.

## A good Wine

Is there a manual published to get people started with Wine, so we can write our own install scripts? There's an interesting point about Microsoft in the world community, which gives free software to open source repositories that Microsoft takes for free. Isn't it about time that *Linux Format* published an article on the predatory Microsoft in the open source community? It owns GitHub and that's not good, is it? For example, I came across a post on [linuxquestions.org](https://linuxquestions.org) about Microsoft *Visual Studio Code* telemetry monitoring what users do with it.

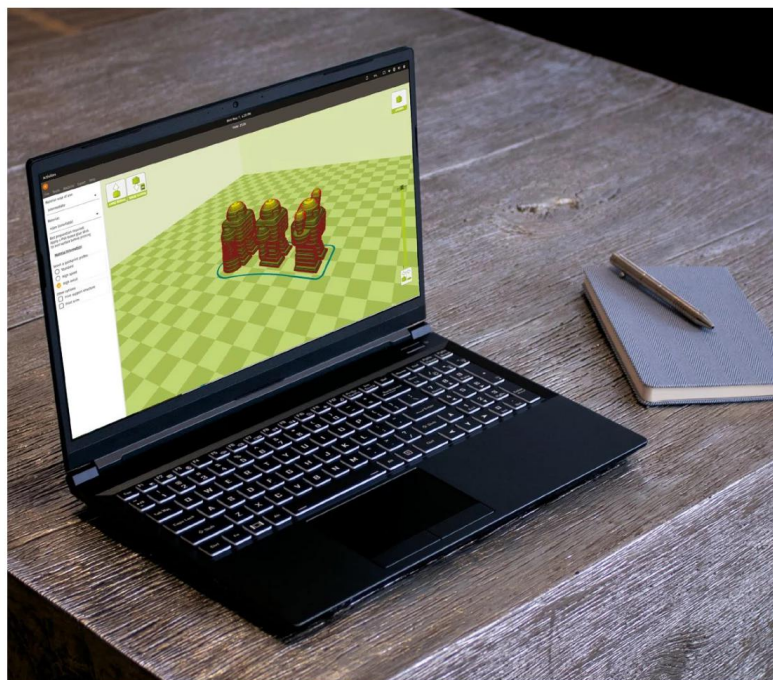
**Clint**

Neil says...

I don't think you write install scripts for Wine. You might be getting mixed up with *WineTricks* or *PlayOnLinux*. The *Wine* FAQ (<https://wiki.winehq.org/FAQ>) is comprehensive, while the *PlayOnLinux* FAQ has a section on writing scripts: [http://wiki.playonlinux.com/index.php/Main\\_Page](http://wiki.playonlinux.com/index.php/Main_Page).

Microsoft is a publicly listed world corporation. It's going to act – as legally it has to – in both its own and its shareholders' best interests, which largely means making money. As a business it's now embracing open source development, extending Linux into its Windows ecosystem and there's a third step, too: enjoying the profits this move brings.

As for the *Visual Code* issue, that's the standard telemetry many companies use to track software usage and improve it. I'm not sure what the controversy is here. There's also a version with this removed (<https://vscodium.com>). But as someone points out there are plenty of code editors for Linux!





## » LETTER OF THE MONTH

### Linux everywhere

I'm a newbie to Linux. Even though I found and installed Ubuntu on a VirtualBox a few years ago, I never really had a play once it was up and running.

Just before the lockdown a colleague of mine mentioned *Plex* to me, which he had running on a Raspberry Pi in his loft. Thanks to my work I had an old HP PRO3400 i3 tower – one that was fitted with a hard drive. I found some RAM and took it home. I already have a small NAS at home that has some entertainment

and photos on it so I decided to give *Plex* a go. Mint was installed, then *Remmina*, as apparently that's a solid VNC/RDP client for Linux and managed to get a VNC connection. Then I installed *Plex*.

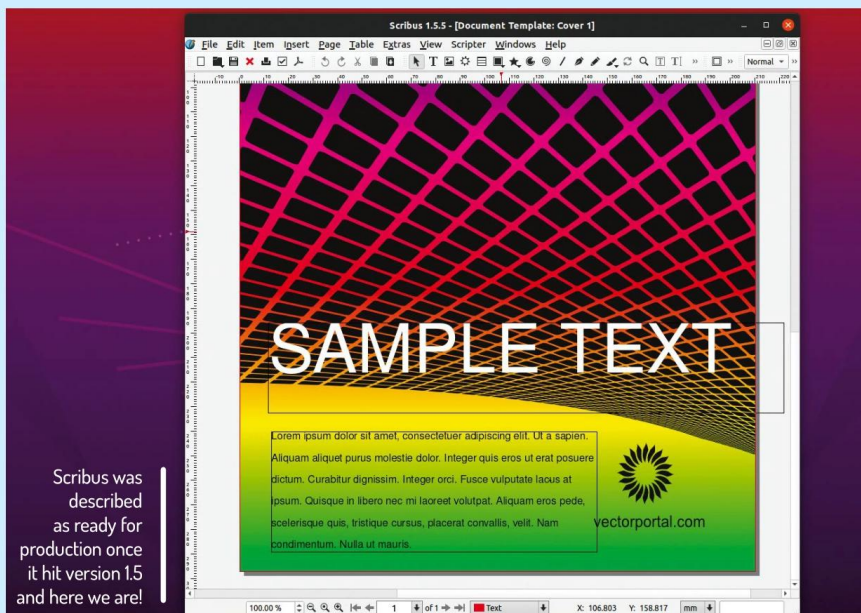
I can say I'm a convert. I love Macs: they work, they work and then they work some more. Windows 10 got me interested in Windows again, but now that I've found Linux, both Mac and Windows will slowly be replaced wherever I can.

A question: I work in reprographics. At this point in time I can't convert my

workspace into Linux – the applications we use aren't available – but I'd like to know what are the available options for replacing such things as Adobe Creative Cloud suite?

In my research so far, I've found programs such as *Scribus* to replace *InDesign*. It can even open IDML files from *InDesign*! *GIMP* can replace *Photoshop* I believe? I have *Inkscape* for *Illustrator* but cannot find a substitute for *Acrobat* yet? I handle a lot of PDFs and need to use the Output Preview palette to view separations.

**Richard Hawker, Nottingham.**



Neil says...

Sounds like you're have much fun using Linux – I hope your *Plex* server is running well? I've been a *Plex* user since before it was called *Plex*. It's certainly had some excellent updates, and I ended up buying a life-time licence.

So *Scribus* is able to open *InDesign* documents? I'm genuinely shocked – I'm going to scuttle off and try this... Also, apparently *Scribus* offer a Print Preview/Pre-flight preview where you can separate CMYK plates – would that suit with your line of work?

A number of commercial PDF tools do support Linux as a platform, such as *pdfToolbox* ([www.callassoftware.com/en/products/pdftoolbox](http://www.callassoftware.com/en/products/pdftoolbox)).

Many thanks for writing in Richard – this was really interesting read. Hope you're getting on with Mint!

### Bluetooth levels

After I obtained a new Android phone I noticed that it displayed the battery percentage charge of any attached Bluetooth device.

This started me wondering if the same would be possible on my laptop, because I've often been caught out by the headset announcing battery low and then switching off, or the mouse suddenly stop connecting due to insufficient battery power? If this isn't

available I think many people would appreciate being able to have a visible battery level shown on the screen for attached devices.

**David Cooper**

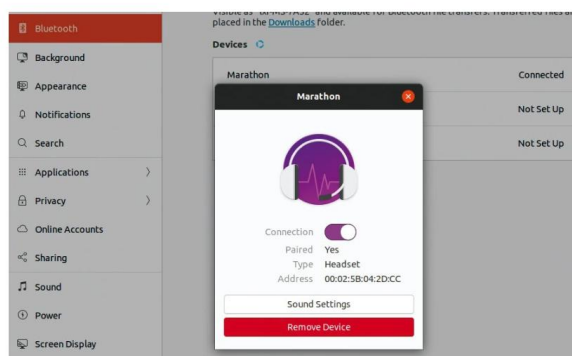
Neil says...

As long as the Bluetooth device supports reporting its battery life level, then the OS should be able to interrogate it for the battery level, but only if your OS has implemented a way, which largely doesn't seem to be the case with many Linux distros. You can try this from the terminal:

```
upower --dump
```

Your device might be listed with its battery status, but it didn't work for us.

Apparently under the Gnome desktop battery level information should be displayed under the Settings>Power tab. However, the version of Gnome in Ubuntu 20.04 doesn't appear to have this, nor does Fedora 32. This could be an issue with our headset, though. Getting the information out of the Dbus system is far from trivial, though – by this we mean it's not a simple single command line solution. **LXF**



■ Ubuntu has no simple way to check the battery level of a Bluetooth device.

### WRITE TO US

Do you have a burning Linux-related issue that you want to discuss? Write to us at [Linux.Format@futurenet.com](mailto:Linux.Format@futurenet.com), Future Publishing, Quay House, The Ambury, Bath, BA1 1UA or email [lxf.letters@futurenet.com](mailto:lxf.letters@futurenet.com).



# EVERYTHING YOU NEED TO KNOW TO GET STARTED WITH UBUNTU LINUX

Packed with easy-to-follow guides and expert advice from the makers of Linux Format magazine, Ubuntu: The Complete Guide is an essential handbook for anyone who wants to get started with the operating system



FUTURE



Ordering is easy. Go online at:

[www.myfavouritemagazines.co.uk](http://www.myfavouritemagazines.co.uk)

Or get it from selected supermarkets & newsagents



## AMD Ryzen 3 3300X

Is this the Goldilocks chip? Well, **Zak Storey** did catch it stealing porridge from Intel, so the moniker might stick...

### SPECS

**Socket:** AM4  
**Type:** 64-bit  
**Arch:** Zen 2  
**Process:** TSMC 7nm  
**Cores:** Four  
**Threads:** Eight  
**Clock:** 3.8GHz (4.3GHz turbo)  
**Cache:** 16MB L3, 2MB L2, 256KB L1  
**Mem:** Two-channel, DDR4 3,200MHz DDR4-2933, two channels  
**PCIe:** x16 v4.0  
**GPU:** N/A  
**TDP:** 65W

**W**e always enjoy looking back at the past with processors, because it gives us a good idea as to what the future might hold, and just how far we've come in such a short time. And with the Ryzen 3 3300X, it's absolutely no exception.

Back in the days of Kaby Lake, a quad-core processor with multi-threading as standard would set you back close to £300, albeit with a slightly higher clock speed. Fast forward three years and that price has halved, and you've gained a bevy of connectivity solutions with it. There's a lot to love here. But this is 12 months on from the initial third-generation Ryzen launch, and we've seen Ryzen 3s before, so why is it so interesting? And why now?

In short, the Ryzen 5 3400G, Ryzen 3 3200G and the Ryzen 3 3100G aren't technically third-generation Ryzen parts. They're in the product stack, sure, launched at the same time as the original third generation, but the reality is these iGPU cousins were meagre imitators, harbouring that pesky 12nm+ Zen architecture we saw with the launch of AMD's second generation. These chips lacked PCIe 4.0 support, and perhaps worst of all only came with eight PCIe 3.0 graphics lanes direct to the CPU itself (the other eight being reserved for the onboard iGPU).

### Do the Core i's still have it?

These were processors designed with the HTPC and home-server markets in mind, where more often than not those PCIe lanes would be taken up with RAID cards or audio solutions instead of graphics. For AMD, the solution was simple for anyone serious about gaming on a budget: get a Ryzen 5 3600. But this left a hole in the market, one Intel filled with its high-performing Core i5s and i3s, all of which featured the full-fat x16 PCIe 3.0 lanes and strong single-core performance. Fast forward a year and at last we see a competitor, in the form of this fruitful Ryzen 3 3300X, in all of its 7nm glory.

So then, is it any good? Well, yes, darn good in fact. In *Cinebench R15* we saw this thing clock in an impressive 199 points in the single-core tests, and 1,123 points in multi-core. Throw it into gaming and it pulls off some impressive performance at 1440p, clocking in 72fps in *Total War: Warhammer II*. Not too shabby.

It does lose some points compared to the likes of its older sibling, the Ryzen 5 3600X, but that's only in heavier multi-threaded titles. Interestingly, we saw temps remain resiliently the same between the two chips, both floating around the 66°C mark under load regardless of core count. This is likely because the



When it's not about the professional end, is four cores still enough?

3300X is still utilising the same number of core complexes as the 3600X, albeit with two additional cores disabled within those CCXs.

AMD's Ryzen 3300X is an interesting, perhaps somewhat late addition to its arsenal. It represents an intriguing proposition for those looking to upgrade their ageing quad-core platforms, with access to newer connection standards and more modern motherboards at an affordable price. But is it too little, too late? AMD has confirmed its fourth-generation chips will be with us before 2020 is out. So investing in a processor that's likely going to drop in price within the space of the next few months seems foolhardy. Nonetheless, right now it's a fantastic processor, and ideal for those looking for something for either home office use, or to use in a slim, lightweight gaming machine. **LXF**

### VERDICT

**DEVELOPER:** AMD  
**WEB:** [www.amd.com](http://www.amd.com)  
**PRICE:** £120

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

For not a lot of cash, it offers a tempting solution for those looking to upgrade their ageing rigs, or set up a new one.

» **Rating 9/10**



# Crucial P2 500GB

What's this? Low-cost SATA-breaking speed without the DRAM?  
Sean Webster is impressed, albeit with a few reservations.

## IN BRIEF

**Size:** 500GB  
**Form:** M.2  
**Type:** PCIe v3 x4, NVMe1.3  
**Controller:** Phison E13T  
**Cache:** HMB  
**Mem:** Micron 96L TLC  
**Seq.:** 2,300/940 MBps (r/w)  
**Ran.:** 95,000/215,000 IOPS (r/w)  
**TBW:** 300TB  
**Warranty:** Five years

**O**n the hunt for a new SSD for your system? Then you've probably noticed there are innumerable options. If you've got the dough, you may want to consider one of the newest PCIe 4.0 x4 options, say the Seagate FireCuda 520 or Sabrent's Rocket NVMe 4.0.

Crucial's P2 is the company's latest M.2 NVMe SSD for the low-budget crowd. The 250GB and 500GB capacities are well-suited for basic boot and application requirements; a 1TB option is on the way.

The P2 features Dynamic Write Acceleration, which programs a pool of TLC flash as faster SLC, for faster write performance, after its fills and write performance degrades. The drives come with RAID ECC and Phison's fourth-generation LDPC that has multiple levels of adaptation. As a result, Crucial's P2's endurance ratings, though they are low for TLC SSDs, aren't as offensive as some of the QLC SSDs we've looked at. The company rates the 250GB P2 up to 150TB of writes within its five-year warranty, and adds another 150TB of write endurance with each doubling of capacity.

The P2 interfaces with the host over a PCIe 3.0 x4 link with the Phison E13T, a DRAMless four-channel NVMe 1.3 SSD controller. There are four NAND packages on the 500GB P2, with nine per cent set aside as factory overprovisioning for background tasks such as bad block management and garbage collection. Each package contains two dies of Micron's latest 512Gb 96-Layer TLC flash.

The P2 lacks DRAM, meaning that it doesn't have a fast buffer space for the FTL mapping tables. Instead, the P2, like most modern DRAMless NVMe SSDs, uses NVMe's Host Memory Buffer feature. With it, the SSD can use a few MB of the host system's memory to provide snappier FTL access, which improves the feeling of 'snappiness' when you use the drive. Additionally, the controller supports APST, ASPM, and L1.2 power-saving modes for better power efficiency.

Efficiency and low heat output are some of the strengths of a DRAMless SSD. Crucial's P2 proves



efficient, delivering more MBps-per-watt over Crucial's P1 and the high-end Samsung 970 EVO Plus.

## Quoted vs actual speeds

Under testing the P2's write speed hit 1.8GBps but even at a QD of 32, it couldn't hit the rated 2.3GBps it should have. Shifting to a 1MB block size and sequential read performance hit the rated 'up to' 2.3GBps spec. Random responsiveness measured well: at QD1, the P2 responded quicker than the Samsung 970 EVO Plus, exceeding it with 14,019/58,092 read/write IOPS.

Crucial's P2 outperforms the P1 massively in heavy write workloads. After writing 24GB of data to Crucial's P2 at a rate of 1.85GBps, the dynamic write cache filled and write performance degraded to an average of 450MBps. It lags high-end competition, but it offers competitive write performance in its class. The write cache also recovers quickly after a minute of idle time.

The P2's single-sided form factor can slide into almost any ultra-thin M.2 NVMe device, and it sips very little power. With a long five-year warranty and NVMe performance, it scores an easy win over any of the SATA SSDs, too. The same goes for the P2 against the P1. While Crucial's older P1 delivers snappier performance under light loads, its 64-Layer QLC NAND flash holds it back during mixed and write-heavy tasks. Crucial's P2 delivers more consistent sustained performance. And, with the increased endurance rating over the P1, we give the win to the P2 as an overall victor. **LXF**



The P2's four TLC 3D NAND flash chips are paired with a third-party Phison PS5013-E13T controller.

## VERDICT

**DEVELOPER:** Crucial  
**WEB:** [www.crucial.com](http://www.crucial.com)  
**PRICE:** £64

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

A solid performance only outdone by the WD Blue SN550, but its low power and warranty still make this one to consider.

» Rating **7/10**



# EndeavourOS 715.2020

Just as HMS Endeavour explored the unknown southern lands, so **Jonni Bidwell** uses EndeavourOS to explore Arch...

## IN BRIEF

An intermediate, rolling release, distro for anyone wanting a taste of the cutting edge. Can be installed with up to nine desktops. An ARM edition is in the works, too. See also: Manjaro, ArchLinux and Fedora.

**E**ndeavourOS aims to be a spiritual successor to Antergos, the Arch-based project that disbanded last year. Antergos was popular because it was one of the first “easy” Arch-based distros, giving users a friendly installer (rather than command line trial by fire) and stylish desktop (without having to put all the bits there yourself).

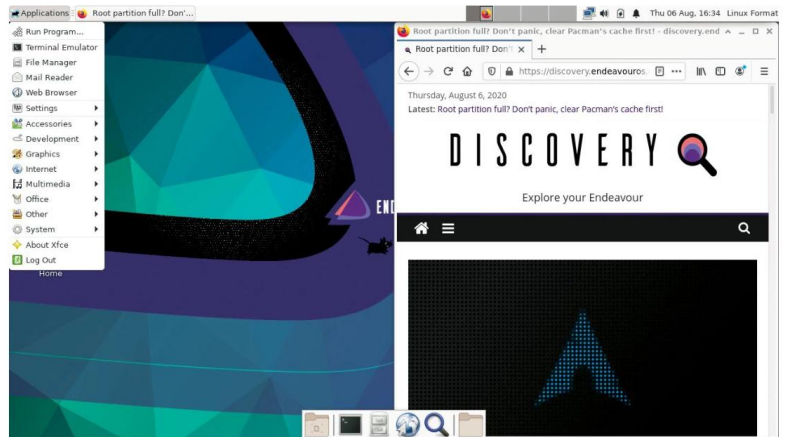
A lot of effort has gone into making GPU setup as painless as possible, particularly for users of Nvidia hardware. The live medium has an option to boot using the proprietary driver, and there’s a helpful documentation page that covers tasks such as using the DKMS script (inherited from Antergos) to build the module on either legacy cards or custom kernels.

The welcome screen offers a choice of offline and online installers. If you opt for offline you’ll get an Xfce desktop, otherwise you can choose from eight others (including i3!). There’s also a helpful option to install printer support. There’s the option to run *Gparted* prior to the Calamares-based installer (if you have advanced partitioning needs), then you can choose manual partitioning to stipulate mount points. If you opt for the Erase disk option then it can set up a swap partition with hibernation support. This is a nice touch for laptops.

Also of interest for laptop users is the documentation’s thorough section on getting Nvidia’s Optimus hybrid graphics working with Bumblebee. During installation, we appreciated the many nods to the May launch of Endeavour, including the “Houston, we don’t have a problem” reassurance that the progress bar isn’t stuck. Once it had schlepped to 100 per cent and we rebooted, we found a well-put-together Xfce desktop, albeit without the insta-searchable Whisper applications menu. The welcome application has links to documentation, post-install tasks and the ability to add popular apps (and indeed the *akm* tool if you want to try different kernel flavours).

Endeavour may not be quite as slick as Manjaro, but has the advantage of being built from the official Arch packages. So you have unfettered access to the official repos, and are less likely to run into weird issues trying to build stuff from the positively brimming Arch User Repository (AUR).

The installed system occupied about 5GB, and in keeping with the Arch way is slightly spartan. It was good to see the *Parole* media player included. If you use RSA SecurID tokens, then the included *stoken* utility can be used to generate token codes. This seems a slightly



The Endeavour community maintains the Discovery website, where you'll find Arch hints, tips and tricks, along with news on upcoming free software.

random inclusion and is most likely a side-effect of bundling Cisco VPN support, also of limited interest.

## Pacman, Yay!

Gnome Packages is there for graphical package management, but you may prefer to use Arch’s *Pacman* from the command line (or indeed any other ALPM-compatible package manager). *Yay!* is also included for AUR packages. You get notifications from the Arch Linux news pages too, which might save you some head scratching if manual intervention is required for any pending updates. There’s also a log tool, which will help you (or the community) diagnose any breakages.

Two Xfce themes (if you count the default one) are included, and you can switch between them from the welcome app. If you want a slightly more novice-friendly, modern experience go with the Gnome or KDE versions.

Endeavour feels a bit rough around the edges in places, especially in its helper tools that like to pop open terminal windows all over the place. The intention behind those tools is good though, and once you’ve learned your way around you can easily disarm them. **LXF**

## VERDICT

**DEVELOPER:** Endeavour team

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

**LICENCE:** Various

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

All in all, a fine introduction to Arch, and great for people wanting to build a bloat-free system.

» **Rating 7/10**



# BunsenLabs Lithium

Forever grunge **Jonni Bidwell** is pleased to find a distribution that’s almost as good as the Nirvana song. Almost...

IN BRIEF

A lightweight Debian-based distribution that leverages OpenBox, Tint2 and Compton alchemy to create a stylish and speedy desktop. Good for aspiring hackers and purveyors of ye olde hardware. See also: ArchLabs, antiX, SparkyLinux, Regolith.

Sooner or later you become fed up of seemingly bloated desktop environments and turn to something lighter, be it OpenBox or i3. These are great – a gift in fact – but the learning curve is steep and a considerable amount of configuration may be required to get them into a state that meets your needs and desires. Even then, there’s the temptation to delve into further rabbit holes, be they compositing and transparency effects, or setting up a custom applications menu. And let’s not even talk about setting up Conky perfectly. Now imagine if a distro came with all of these set up out of the box...

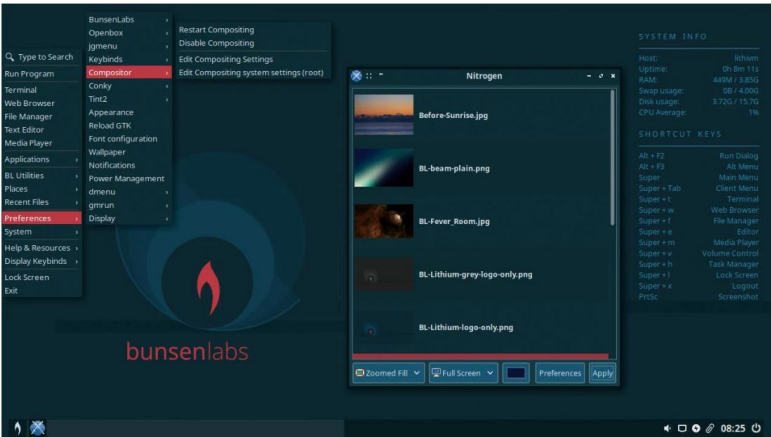
Well good news, BunsenLabs does. And the new release is named after everyone’s favourite rare earth metal. It’s based on Debian Stable, so users searching for the latest versions of everything may want to look elsewhere. But for everyone else there’s a lot to like here. New in this release is the adoption of jgmenu, as opposed to Openbox’s native menu. The new system is configurable and works with Openbox pipe menus too, so if you’ve already created the perfect menu system there, then it should be easy to migrate.

Booting into a fresh install, you’ll be prompted to run an optional post-install script. This first warns you of the perils of adding Ubuntu PPAs or badly backported Debian packages, then updates your system. Next, it gives you the option of enabling the official Debian backports repo or Bunsenlabs own backports – again, warning you to be careful of these packages. Then it offers to set up Bluetooth, Java, Flash and Dropbox. It can optionally install Bunsen’s meta packages for such things as Version Control Systems, web development and building Debian packages.

Mean and moody

With that script out of the way you can admire the full extent of BunsenLab’s style. The moody, dark theme is consistent across the desktop, Tint2 panel and Thunar file manager. If you browse in here you might notice the custom Papirus icon set, too. With the initial install coming in at under 3.5GB, there’s not much in the default install, but we were pleased to see the Geany text editor (which is almost an IDE) and you have Firefox and LibreOffice, which covers a good chunk of most people’s computing needs.

Where BunsenLabs excels is through showing you where all the desktop guts are configured, and offering easy menu shortcuts to do so. No more must you wrestle to open another terminal to restart Conky. While



Bunsenlabs: keeping Conky configurations and cascading menus cool since CrunchBang Linux’s unfortunate demise in 2015.

the list of keyboard shortcuts in the latter is helpful to get started, sooner or later you’ll want to replace all that with Conky wizardry and widgetry. If you do need help, there’s a friendly community on the forums to assist you. And a whole subforum dedicated to using Conky to give you weather and lunar phase information on your desktop, should you need it. There’s no official documentation besides the release notes and Getting Started sections in the forums, but the excellent Debian docs should at least point you in the right direction if you get stuck.

As you may suspect, the Openbox/jgmenu desktop is snappy and responsive. The applications (besides LibreOffice and Firefox) are lightweight, but still easy to use. Xfce’s aforementioned Thunar is the default file manager and NetworkManager manages connectivity, so you won’t need to get your hands dirty doing that. VLC will cover your media playing requirements.

BunsenLabs is one of an ever-decreasing number of distros to offer a 32-bit version (like Debian 10 it’s supported until 2024), so if you’re looking for something new for your old machinery, this is very probably it. **LXF**

VERDICT

DEVELOPER: BunsenLabs team  
WEB: www.bunsenlabs.org  
LICENCE: Various

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

A flexible, stable, well-put together distribution that’s well worth spending some time with.

» Rating 8/10



# Tropico 6

Management are terrified of the new *Linux Format* workers' council, but it's just **Phil Savage** playing at communism under a hot sun.

## SPECS

**OS:** Ubuntu 16.04 LTS, Steam OS  
**CPU:** AMD A10 7850K, Intel Core i3-2000  
**Mem:** 8GB  
**GPU:** Radeon HD 7870, Geforce GTX 750, 2GB VRAM

**T**ropico 6 is a great game for people watching. It's a satirical city builder in which every one of the citizens of your banana republic is simulated. You place a mine. You watch as a construction crew makes its way over to the building site. You watch as the newly constructed mine's employees start digging for gold or coal or uranium or whatever. And you watch as teamsters come to take the raw materials to a factory for processing.

When it's all going well, there's a calming rhythm to the bustle of your island. When it's not, you find yourself scouring around the map, trying to diagnose problems. The focused, individualised simulation means that small inefficiencies can balloon into big problems, and the behaviour of your citizens feeds back into wider systems in interesting ways. That's why – while there's a sandbox mode with plenty of different islands and options – often the *Tropico* series is at its best during the campaign missions, where specific requirements force you to adapt.

## Curveball missions

In sandbox, you can go slow, sensibly growing your island, diligently pursuing new financial ventures, effectively placating political factions and superpowers. You have the space and freedom to effectively manage your growth as you progress through the different eras. But the missions – presented as an anthology of past adventures, narrated by your trusty aide Penultimo – throw in entertaining curveballs to overcome.

Each focuses on a different aspect of the game, be it the spread of propaganda, the balancing act of international relations, or the benefits of light piracy. The latter is one of the most entertaining. Starting on an island with virtually no natural resources, you're required to pillage raw materials to then manufacture into more profitable goods. The raid system is a



Developing a space programme increases the happiness of all Tropico citizens. Civic pride trumps any misgivings about uncovering the secrets of the Solar System purely for profit.

that aren't supported by local crops is a meaningful twist on a standard campaign.

If there's a downside to raids, it's that there's no real downside. Foreign powers have traditionally taken a dim view to piracy, but in this, a game that specifically pokes fun at international relations, it just doesn't come up.

Tonally, *Tropico* is almost too broad and bawdy to be considered satire, but the over-the-top absurdity does lead to some fun mission requests that feed comedy into mechanics. In one, for instance, the communists instructs you to dismantle religion, banks and mansions. This leads to outrage from the capitalists. The only thing that can placate them? Building a golf course.

*Tropico 6* was developed by new series stewards Limbic Entertainment, but you'd be hard pressed to know by just playing the game. It takes *Tropico 5*'s era system, reintroduces *Tropico 4*'s political speeches and work modes, and adds in a few new features designed to complicate supply chains and diversify systems. It looks a bit nicer – the *Tropico* series has always been pretty – and the archipelagoes, bridges and tunnels add a few neat wrinkles to construction. That's about it. **LXF**

Build this banana republic into a force to be reckoned with – and get rich, too!



powerful new tool, essentially gifting a regular trickle of goods, immigrants and, in later eras, beneficial propaganda and even falsified tourist reviews. Having to create supply chains

## VERDICT

**DEVELOPER:** Kalypso Media  
**WEB:** [www.kalypsomedia.com/uk/games/tropico-6](http://www.kalypsomedia.com/uk/games/tropico-6)  
**PRICE:** £46

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

An entertaining but unambitious sequel that collects up the best features of previous games and adds some new twists.

» **Rating 8/10**



**SUBSCRIBE** Save money today!

# SUBSCRIBE

Sign up today and get your

## Eclipse 3-Way USB Charger

A circular, wood-grain textured USB charger with a silver rim. It is shown charging three devices: an iPad on the left, an iPhone on the right, and a laptop at the bottom. Each device's screen shows a 100% battery level. A yellow circular callout above the charger says 'YOUR GIFT! WORTH £64.95 Don't miss out, subscribe now!'. To the right of the charger, a list of product features is shown. Below the charger, a paragraph describes its features. The entire scene is set against a light grey background with a blue border.

**YOUR GIFT!**  
**WORTH £64.95**  
Don't miss out, subscribe now!

**Product features**

- » Holds 6m of cable
- » High-speed charging
- » Smart-IC Technology
- » Three devices at once
- » No more messy cables

Charge multiple devices at maximum speed with the **Eclipse 3-Way USB Charger**. This easy-on-the-eye charger is a three-port USB charging hub that keeps 6m of cables neatly wrapped inside when not in use, while providing neat pop-up access with a gentle press of its top.

# SUBSCRIBE NOW!

[www.myfavouritemagazines.co.uk/lin/eclipse20](http://www.myfavouritemagazines.co.uk/lin/eclipse20)

Call: 0344 848 2852



FREE DVD! FEDORA 24 &amp; VOYAGER

**LINUX**  
FORMAT

Get into Linux today!

**HACK LINUX!**

Get under the hood and unlock even more power!

**Fedora 24**  
The cutting-edge distro  
» The latest kernel 4.5  
» GNOME 3.20 desktop  
» Flatpak release system  
» Wayland display stack**DON'T MISS!**  
Now with 5 years of Linux User & Developer issues**65** Pages of tutorials and features  
» Pilot your own Pi drone  
» Take control with GPS  
» Monitor Ubuntu and more!**» PLUS:** Exclusive access<sup>1</sup> to the *Linux Format* subs area!1,000s of DRM-free PDF back issues and articles! Get **instant access** back to issue 66 (May 2005) with tutorials, interviews, features and reviews. At [linuxformat.com](http://linuxformat.com)<sup>1</sup> Only available to MyFavouriteMagazines.co.uk subscribers.**» CHOOSE YOUR PACKAGE!****6-MONTH PRINT ONLY****Only £32.50**

6 month print by Direct Debit (NO GIFT)

**ANNUAL PRINT ONLY****Only £67.00**

Annual print by Direct Debit

**ANNUAL PRINT AND DIGITAL****Only £72.00**

Annual print and digital by Direct Debit

This offer is only available for new UK subscribers. Gift is subject to availability (MSRP £64.95). Gift colour may vary. Please allow up to 60 days for the delivery of your gift. In the event of stocks being exhausted we reserve the right to replace with items of similar value. You will receive 13 issues in a subscription year. You can cancel your subscription within 14 days of purchase. Payment is non-refundable after the 14-day cancellation period unless exceptional circumstances apply. Your statutory rights are not affected. Prices correct at point of print and subject to change. Full details of the Direct Debit guarantee are available upon request. For full terms and conditions please visit [www.bit.ly/magterms](http://www.bit.ly/magterms). Offer ends 31 October 2020. UK calls will cost the same as other standard fixed line numbers (starting 01 or 02) or are included as part of any inclusive or free minutes allowances (if offered by your phone tariff). Lines are open Tue to Thu 9am to 5pm (British Summer Time).



# Roundup

Linux Kodachi » Septor Linux » Tails  
» TENS » Whonix



**Shashank Sharma**

is a legal Delhi-based master at keeping his privates covered!

## Privacy distributions

Only Batman can give you “clean slate” software, but if it’s anonymity you’re after, **Shashank Sharma** knows of some distributions that can help.

### HOW WE TESTED...

Specialist distributions such as the ones featured in this *Roundup* can’t be tested with the same metric that we would use for desktop distributions. While ease of use is still important, we’re far more interested in what each has to offer, and whether the distros will help you browse the internet, exchange emails, and otherwise interact online without fear of compromising your identity. To that end, we’ll test the five distributions on the protection mechanism they use to mask your online footprint.

A key use for such distros is when working with unfamiliar environments, such as hotels and cafés, so deployment is also important. These distros all use a variety of tools and technologies to keep you anonymous, so proper documentation is also expected from each. Just as important is whether they can keep your data safe and if can they be used as a desktop distribution, too.



**O**ne unexpected outcome of the Covid-19 outbreak is that it’s managed to convince even the most technologically averse people to connect online for a variety of tasks. Everything from initiating funds transfer and filing of taxes, can be done online.

Regardless of your fluency with the internet, it’s only natural to wonder if your online activities are safe from prying eyes. The distributions featured in this *Roundup* enable you to defend your privacy and prevent any inadvertent leaks of sensitive data.

The distros take different approaches and come with their own traits and benefits. Some

rely on routing your web traffic via well-known anonymising networks such as Tor, while others use novel approaches such as security by compartmentalisation. Because anonymity and security tend to go hand in hand, using these distros will help you protect your computer from digital assailants.

As with many other open source projects, niche distributions aren’t beyond dying or unexpectedly becoming dormant. For this *Roundup* we’ve narrowed down distributions that are still in active development, while some alternatives, although not officially dead, are mentioned in the *Also Consider* section at the end of the article.



# Protection mechanism

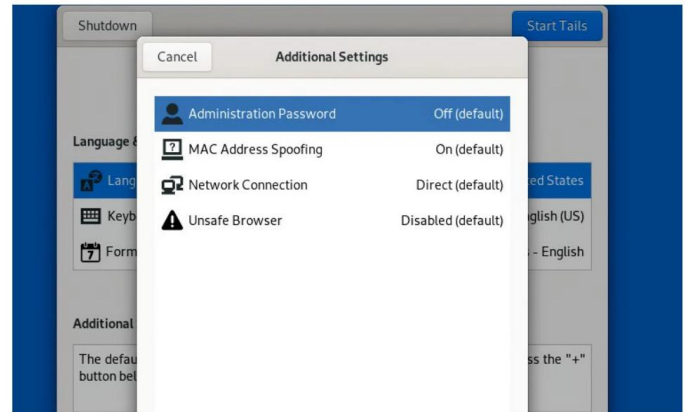
What do they use?

**T**here are any number of tricks utilised by websites to track your online footprints. Everything from what websites you access to the online purchases you make are all tracked and monitored, so as to enable certain websites and portals to provide an online experience that's tailored to your exact tastes. You can shield yourself from some of these nefarious activities by purging the browser cookies and taking various other precautions. The more advanced of which is to use the *Tor Browser* and operate from behind a VPN.

Tails is built around Tor, which is an open network of anonymous servers that attempts to prevent your identification. The distro also includes tools that help set up the network and a web browser with privacy-enhancing extensions. Additionally, Tails includes a couple of useful cryptographic tools to encrypt disks and online communication.

Whonix is built on the concept of security by isolation. The distribution comes in the form of two virtual machines. The idea behind this approach is to isolate the environment you work in from the internet access point. On top of this, Whonix routes all internet traffic through Tor. Thanks to this setup, even if one of the machines is compromised, it would be impossible to uncover your real IP.

Kodachi first routes all connections to the internet through a VPN before passing them to the Tor network. It also includes a collection of tools to easily change identifying information such as the Tor exit country as well as resetting your DNS servers,



There's no getting around the thoroughness of Kodachi, but that shouldn't discourage you from using Whonix, Septor or Tails, which have their own USPs.

deploying your own VPN, and more. The distro also encrypts the connection to the DNS resolver and includes well-known crypto and privacy tools to encrypt all your offline files, emails and instant messaging. When you're done using it, Kodachi also removes traces of its use from the computer during shutdown.

Like the other distros, Septor also routes all internet-bound traffic through the Tor anonymous network. The distro also includes a couple of tools that are designed for use over the Tor network, such as *OnionShare* for file sharing, which you'll find in the other distributions as well.

## VERDICT

LINUX KODACHI	10/10	TENS	5/10
SEPTOR LINUX	9/10	WHONIX	9/10
TAILS	9/10		

Although it's functional, all other distros have more to offer than TENS.

# The unique selling points

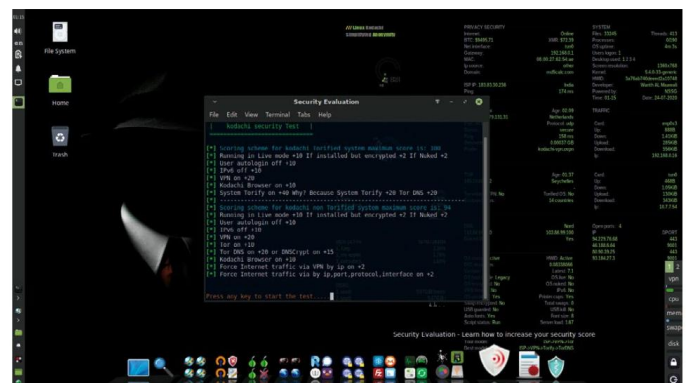
What makes them stand out?

**A**lthough the distributions strive to ensure your online privacy, there's plenty to distinguish them from one another. Linux Kodachi is developed by a professional IT security service provider. The project went into a period of hibernation until 2016, but now ships new releases and updates at regular intervals. With its out-of-the-box software offerings, the distribution towers above the competition. Unlike the others, it recommends running it in a virtual machine.

Septor Linux is still quite new, having debuted in 2018. It produced a single release in 2019, but has already had four releases so far in 2020. Each new release ships updates from the Debian Buster repository, along with updates for various applications.

TENS, which was previously known as Lightweight Portable Security, is updated regularly, typically with quarterly maintenance releases. It's designed, developed and distributed by the United States Department of Defense's Air Force Research Laboratory.

The best feature of Whonix is the amount of documentation on offer, which you can use to fine-tune the distribution and bundled applications and utilities. But it's ill-suited for machines with limited



Linux Kodachi ships a security evaluation utility to help you configure the system for optimum security.

resources because it requires simultaneously running two virtual machines. This will be a tall task for a dual-core machine with only 4GB RAM. Finally, there's Tails, which is one of the best-maintained security distros. Its fast pace of development sometimes results in two new releases in a month.

## VERDICT

LINUX KODACHI	10/10	TENS	9/10
SEPTOR LINUX	9/10	WHONIX	10/10
TAILS	10/10		

All distros come with a variety of unique, useful features, and score highly.



# Ease of use

Do you need special skills before you can use them?

**U**nlike desktop distributions that need to be usable, flexible and attractive so they can appeal to a large section of users, niche distributions such as the ones featured in this *Roundup* have a far greater responsibility: delivering the intended functionality. Everything else must take a back-seat. However, this isn't to say that the distributions can insist users learn how to master juggling bowling balls while blindfolded before they can use the distribution.

Ultimately, these distributions provide the perfect environment to use some specialised software and to safeguard your online presence. But being anonymous while maintaining your privacy on the internet requires compromise. For instance, you can expect slower browsing speeds while the data packets are routed all over the world before finally arriving at your computer. We're looking for a distro that provides a smooth browsing experience without compromising users' safety and security.

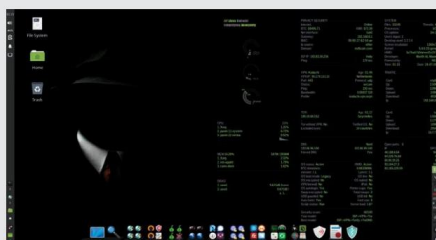
## Linux Kodachi

9/10

The distro boots into a modified Xfce desktop environment that displays details such as status and IP address of the VPN, the Mac address, Tor IP, CPU usage, memory and traffic data on the desktop. Once Kodachi is connected to the internet, you can start the Kodachi VPN which will automatically initiate the Tor connection.

Advanced users can also connect via their own VPN. The myriad software that the distro ships with is neatly packaged into distinct categories on the dock at the bottom of the screen. This includes the VPN tools, security applications and so on.

The Panic Room entry includes various privacy tools such as *USB Guard* to generate new MAC addresses for all interfaces, a tool to wipe RAM, and *Nuke*. The *Nuke* option creates a password for an already encrypted Kodachi installation and as soon as you enter this password, all contents of your installation are destroyed.



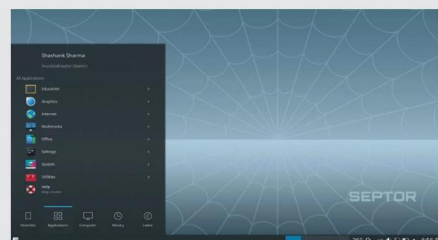
## Septor Linux

7/10

Septor is based on Debian's Testing Branch and ships with a KDE desktop environment. Compared to the other distributions, Septor is slightly sluggish at times, especially when navigating the Applications menu.

While Live-installable distros usually enable you to run the installer from within the live environment, with Septor, your only option is to launch the installer from the boot menu. You can opt for a text- or graphical-installer. It's the only distribution to still feature the graphical front-end for *MAT*, while others now only offer the command-line utility.

Both Septor and Tails have very little to offer outside of their use of Tor to route internet traffic and offer similar set of applications out of the box. Unlike Tails however, it doesn't provide any mechanism to wipe RAM or other protections against forensic recovery.



# Documentation and support

Junk mail and slow Wi-Fi are annoying. So are unanswered questions.

**K**odachi offers a quick FAQ and instructions on how to use and install the distribution, along with some information on usage, but if you're looking for more comprehensive documentation detailing all the software shipped with the distro or how to use it, you'll be disappointed.

Septor Linux fares only slightly better. Its Sourceforge-hosted website is home to a wiki that contains a few text-heavy articles describing some of the software you'll find in the distro. Except for a quick write-up on Tor Birdy and *Privoxy*, there are no user guides, or forums or mailing lists where you can connect with fellow users and seek assistance.

Tails offers in-depth end-user documentation in multiple languages with general information, first steps, commonly asked questions and more. You'll find directions on installation, creating persistent storage, using encryption to protect your data and more. There's even an XMPP chat room, a support mailing list and a form to request features. Users can email queries in English, French, Spanish and Italian, and the website also provides a list of

third-party organisations from around the world that conduct training on how to effectively use the distribution.

Apart from FAQs covering its features and installation, the TENS's website features a quick-start guide as well as a user guide. The latter can also be accessed from the desktop itself.

Whonix won't let you down either with its vast wiki containing detailed documentation split across different sections such as First Steps, Basic and Advanced Security Guides, Anonymous Browsing and more. The distro also offers several support options, including professional support at €200 per hour.

## VERDICT

LINUX KODACHI	4/10	TENS	8/10
SEPTOR LINUX	4/10	WHONIX	10/10
TAILS	10/10		

Kodachi's ease of use acts as its life vest, otherwise the lack of detailed documentation would be fatal.

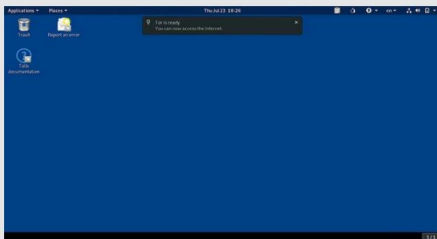


**Tails****8/10**

You can use Tails without making any changes when you boot into the Live environment, but the welcome screen provides the option to tweak certain settings. The welcome screen can be used to specify an administrator password, disable MAC address spoofing and enable the unsafe browser, apart from tweaking the language and region settings.

You can access the Tails documentation using the icon on the desktop, which launches *Tor Browser* and navigates to the online documentation. The Debian-based distro features Gnome desktop, and automatically connects to the Tor network as soon as you log in. You can click the Tor icon in the status bar to view the circuits and streams.

When running through USB, the distro also offers to download and install updates as they become available. The updates are implemented onto the USB drive on the fly.

**TENS****7/10**

On booting, TENS first asks you to agree to the user agreement. It informs you that the distro is based on GPL but is offered without warranty. The distro brings up the desktop once you agree and press Continue. Although the distro uses the XFCE desktop, it's made to resemble Windows XP. Everything from the desktop's layout, complete with the Windows's Start button to the window decorations, has been designed to ape the proprietary OS.

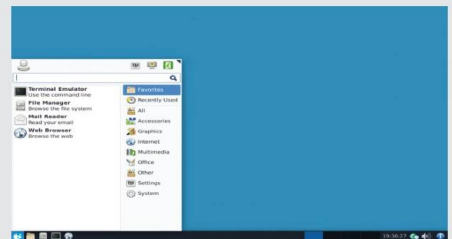
Using TENS is fairly intuitive. One of the hallmarks of the distro is the *Encryption Wizard*. When launched you can drag and drop files and specify the password to lock them. You can also opt to encrypt the files with a certificate file or by generating a secure password to get a stronger passphrase. The tool also asks if you wish to retain the original file, or only the encrypted version. You can also securely delete such sensitive files from the tool.

**Whonix****8/10**

Whonix ships as a pair of Debian-based virtual appliances that you must run simultaneously in two separate virtual machines, where one is your workstations while the other acts as a gateway. The obvious downside to this approach is the hardware requirements. The machines by default are configured to utilise three CPU processors each, so if you're on a quad-core or lesser machine, then you'll have to tweak the configuration accordingly.

The *iptables* rules on the Whonix-Workstation force it to only connect to the virtual internet LAN and redirect all traffic to the Whonix Gateway. This scheme prevents applications from ever knowing the user's real IP address or accessing any information on their physical hardware.

The virtual machines occasionally behaved erratically and failed to launch some programs. The problem was resolved after restarting the virtual machines.



## Deployment flexibility

Can you anchor them to the disk? And should you?

**M**ost of the distributions featured in this *Roundup* recommend that users not install the distribution to disk for maximum protection against potential risks. While you can try and use all of them using virtualisation software such as *Oracle Virtual Box*, some distros also recommend against this practice.

The exception is Whonix. The quickest way to deploy the distribution is to download the virtual machines. You're expected to then boot into these machines, which function as any other installed distribution.

TENS is also peculiar in its approach in that it doesn't support installation to disk, or USB stick for that matter. Instead, it expects users to run its live environment.

You also can't anchor Tails to the hard disk on your machine. The installer utility only permits you to install it to a USB device or a SD card. If you've configured the live environment, such as for using Wi-Fi, you can even clone the current environment with the installer utility.

Based on Xubuntu, Kodachi utilises its Ubiquity installer to full effect to help you install the distribution to the disk. It provides the option to run the installer in online or offline mode through two distinct icons on the desktop itself. If you choose the latter option, the distro shuts down all internet traffic and takes you offline before starting the installation process. Due to the vast number of packages on offer, the distribution requires at least 17GB of free space on the hard disk.

With Septor Linux, the only way to achieve a persistence mode is through installing the distribution. Its graphical and text installers can be accessed only from the Grub boot menu.

**VERDICT**

<b>LINUX KODACHI</b>	<b>6/10</b>	<b>TENS</b>	<b>N/A</b>
<b>SEPTOR LINUX</b>	<b>5/10</b>	<b>WHONIX</b>	<b>N/A</b>
<b>TAILS</b>	<b>7/10</b>		

While Whonix is at least configurable in its approach, TENS isn't.



# Bundled applications

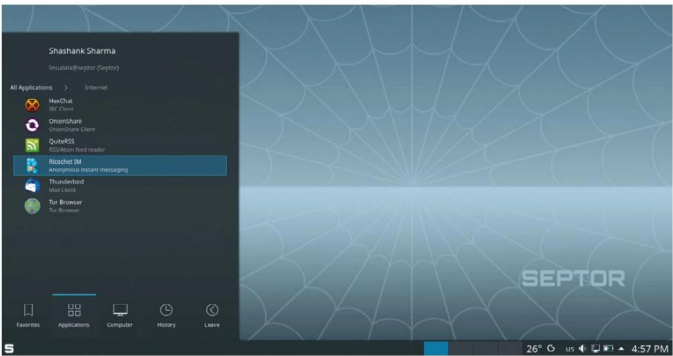
Can they readily replace your daily desktop distribution?

**T**ails ships with a number of desktop applications such as *LibreOffice*, *GIMP*, *Pidgin*, *Inkscape*, *Audacity* and *Thunderbird*. You can also utilise the *Synaptic Package Manager* to flesh out the Gnome-powered distro. Any packages you choose to install aren't made available at subsequent reboots, unless you configure persistent storage.

Whonix has only a limited applications on offer. You get *VLC* to play multimedia files, an image viewer, a PDF reader and *qTox* IM client. You can install additional software using **apt-get**, but the distro doesn't ship a graphical package manager.

You get *LibreOffice*, *VMWare View*, *Totem Movie Player*, an image viewer and *Evince* PDF reader as part of TENS's Deluxe Edition. These are complemented by *Firefox*, *Thunderbird*, *Pidgin* and a desktop sharing utility. The project also produces a regular release, which lacks *LibreOffice* and other programs. Unlike the other distros, TENS doesn't support installation of more software and doesn't ship with a package manager to flesh out the distro.

Septor makes it possible for you to install additional software using *Synaptic Package Manager*. But until you do, you'll have to make do with *GIMP*, *Gwenview*, *Thunderbird*, *Ricochet* which utilises



With the exception of TENS, all the distros use apt-get for package management.

Tor network for anonymous instant messaging, *LibreOffice*, *VLC* and more.

Linux Kodachi scores over the others and includes the widest set of applications. The distro caters to all kinds of users and includes *VLC*, *Audacity* and *LibreOffice*. Also on offer are enthusiast applications such as *GIMP*, *OpenShot* video editor, *Gmerlin Player/Mixer/Transcoder* and more. You can use the *Catfish* file search for finding files. You can also install additional software from the repositories using the *Synaptic Package Manager*.

VERDICT			
LINUX KODACHI	10/10	TENS	8/10
SEPTOR LINUX	8/10	WHONIX	7/10
TAILS	8/10		
Kodachi is the most usable distribution straight out of the box.			

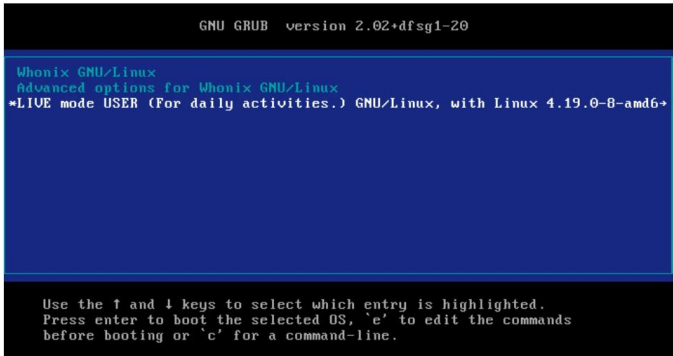
# Security applications

How do these privacy distributions protect you and your data?

**A**s well as offering a variety of tools such as *USB Guard* out of the box, Kodachi takes many other steps to bolster security. The distro uses *AppArmor* for application isolation. You can use *KeePassXC* for managing passwords. You can also safeguard data within encrypted volumes using *ZuluCrypt* or *VeraCrypt*. It also features various tools for hiding traces of use from the system such as *Metadata Anonymization Toolkit (MAT)*, for zapping metadata information from files, *BleachBit* and more. You'll find all these and more applications and utilities on the dock, at the bottom of the screen.

Tails is primarily funded by the Tor project and it too utilises *AppArmor* for application isolation. You'll also find a password generator to help you come up with strong passwords, and *KeePassXC* for securely storing them. Like the other distros, it also provides *MAT* out of the box. As an added precaution, the administrative password, which is needed to perform tasks such as installing software and executing commands which require root privileges is disabled by default in the distribution. You can set one up from the Welcome Screen, but not from within the Gnome desktop. The distro is also configured to wipe your RAM on reboot or shutdown to protect against forensic recovery techniques.

If you want the same capabilities with Whonix, you must boot into its Live mode, instead of booting into the virtual disks. This



Use Whonix in Live mode if you don't want persistent storage.

presents additional security as all changes made to the system, such as newly installed software, websites that you've browsed and downloaded content is automatically flushed as soon as you restart or shutdown.

TENS features the *Encryption Wizard* which, like the distribution itself, is created by the US Air Force Research Laboratory to encrypt documents and directories. It uses 128- or 256-bit AES encryption, and supports compression of encrypted archives. You can also enable *DNSCrypt* for additional security.

Besides the Tor tools, Septor also has a few other privacy and security-enhancing programs such as the *Sweeper* utility to clear the cache and temp files, *MAT*, and the *VeraCrypt* encryption utility.

VERDICT			
LINUX KODACHI	9/10	TENS	6/10
SEPTOR LINUX	7/10	WHONIX	8/10
TAILS	8/10		
TENS is the only distribution to not offer MAT out of the box.			



# Privacy distributions

## The Verdict

**T**he past 20 years have seen a tectonic shift in people's perception of invasion of privacy. While there have long been stories about state surveillance, it wasn't until large corporations began collecting user information that people became alarmed. Even as simple a task as looking up Father's Day gift ideas online can tailor the advertisements you see for weeks to come. If you find this behaviour worrisome and wish to ensure your online activities remain hidden and unknown, you must use tools such as the Tor network and a VPN to mask your online presence. But that's hardly enough, because there's still plenty of trace evidence on your hard disks and other physical components such as RAM to divulge secrets.

This is where anonymising distros come in. While you can easily set up Tor on your existing Linux distribution, it'll take far more work to completely guard your online activities, hence the need for anonymising distributions.

TENS has little of value to offer when compared to the other distros. Worse still is its origins in the United States Department of Defence's Air Force Research Laboratory. Couple that with the inability to flesh the distro with additional software and it becomes apparent why TENS ranks last.

We quite enjoyed working with Septor, which isn't all that different from Tails. The one major difference, however, is also why the distro didn't make the podium: the lack of documentation and support avenues.

Tails outperformed Whonix on our test machines, and although both are popular and well documented, we couldn't possibly let Whonix outrank Tails with its erratic behaviour.

Just as Septor Linux, the worst aspect of Linux Kodachi is the lack of documentation and support options, which is why we can already hear you protesting over our awarding of Kodachi the top spot. But our reasoning will become obvious once you use the distro. In addition to the sheer number of programs on offer, our decision was influenced by the ease of use and the everything-works nature of the distro. The latter served as a life vest and kept the distro from sinking past the podium. **LXF**



### 1st **Linux Kodachi** **9/10**

**Web:** <https://digi77.com/linux-kodachi> **Licence:** Various  
**Version:** 7.1

Ideal for anyone looking to be safe while online.

### 2nd **Tails** **8/10**

**Web:** <https://tails.boum.org> **Licence:** GPL v3  
**Version:** 4.4.8

Not as versatile as Kodachi in terms of offerings, but flawless otherwise.

### 3rd **Whonix** **8/10**

**Web:** [www.whonix.org](http://www.whonix.org) **Licence:** GPL and others  
**Version:** 15.0.1.3.4

Excellent for experienced campaigners with modern hardware.

### 4th **Septor Linux** **7/10**

**Web:** <https://septor.sourceforge.io> **Licence:** GPLv3  
**Version:** 2020.3

Documentation and lack of deployment flexibility are a problem.

### 5th **TENS** **5/10**

**Web:** [www.spi.dod.mil/lipose.htm](http://www.spi.dod.mil/lipose.htm) **Licence:** GPL and others  
**Version:** 3.0.1

Nothing of particular interest here.

## » ALSO CONSIDER

One distribution that we were eager to feature in this month's *Roundup* but ultimately decided against doing so is Mofo Linux. Based on Ubuntu, the Mate-powered distro is chock-full of applications, and could give Kodachi a run for its money.

In addition to Tor, Mofo Linux also provides users with the option of using i2P and Freenet. It also offers tools such as *Lantern* proxy and more out of the box. Unfortunately, the distro hasn't seen a new release since February 2019, with no

news on upcoming releases. This lack of development held it back from featuring on our test machines.

Discreete Linux is another option that didn't make the cut for the same reason. It started as Ubuntu Privacy Remix way back in 2007, but hasn't seen a new release in the past several years. If you want a single distro that's designed for pen-testing, privacy, digital forensics, reverse engineering and software development, you should consider using Parrot Security.



# BACK UP LIKE CLOCKWORK!

**Jonni Bidwell** has a plan – several plans, in fact – to keep his ever-growing repository of data ticking over nicely.



**T**here are two types of people in this world: those who back up their files and those who have never experienced data loss. In a world that's becoming increasingly cloudy, backing up files might seem like an old-fashioned sort of a habit. Much like wearing a watch or writing cheques (...that your body can't cash? – Ed).

If your Android phone dies (and you opted in to connecting everything to your Google Account) then you can restore your contacts, messages and apps effortlessly on a new device. If you work in a corporate environment, then your IT

team can (with a little more work) probably do a similar thing if your work machine dies. You might lose a few hours' work (depending on when and where you last saved) or some tiny configuration details, but starting from a base image applications can be re-installed and your user account connected in a matter of minutes, not days. But if your home PC fails, or even if you accidentally delete some files, recovery can prove much more complicated.

Unless, that is, you have a solid backup regime, which hopefully this feature will inspire. Through automated, incremental backups we'll aim to make

backups less chore-like. We'll cover snapshots so that your OS can be restored in a few clicks. We'll look at using multiple drives (RAID) to protect against hardware failure, paving the way for ill-considered jokes about redundancy. We'll look at the excellent *Duplicati* tool, which can integrate with cloud storage services as well as a VPS, or indeed a real server, under your control. If you prefer to do things manually, then we'll also look at the command line program *Duplicity*, which uses *Rsync* to send encrypted tar archives to the clouds. And because we like choice (and alliteration) we'll look at making this easy with the *Déjà Dup* frontend.



# Pre-backup considerations

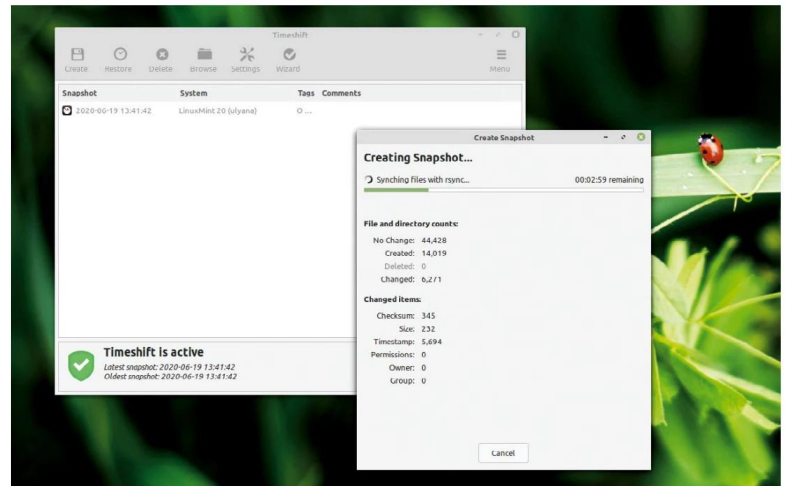
A quick look at the why, what and where of a strong back-up plan.

**O**ver the second quarter of 2019, a survey by cloud storage firm Backblaze recorded a 1.8 per cent annualised failure rate over 108,461 hard drives in its service. So extrapolating wildly, we might say, in lay speak, that there's a 1.8 per cent chance that an 'average' disk will fail within a year. Those are enterprise-class drives as well, so the odds are only going to be worse for consumer hardware.

A significant obstacle to having a watertight back-up regime is how disorganised we are with our files in the first place. One upside to the Gnome desktop abolishing the right to store files on the desktop is that it obviates the terrible habit of covering your lovely desktop wallpaper with so many icons and partial file names. It belongs to the same organisational scheme as putting everything on the floor so you know where everything is. If, on the other hand, you keep all your photos (including those from your phone) in one folder, important documents in another, large dataset that can't be easily replaced on their own drive, and so on, then it's much easier to put together an efficient back-up strategy.

It's also useful to segregate system and user file backups. In many cases, backing up your operating system is less important than your own data. You can reinstall Linux in five minutes, so it really depends how much manual configuration you've done.

Data stored on Dropbox, iCloud, Google Drive or Amazon S3 (or any other cloud storage service you'd care to name) is unlikely to just vanish in the way that data stored locally would if your hard drive goes kaput (as long as you keep paying your bills, of course). Yet these services suffer hard drive (or solid state, tape or Blu-ray) failures all the time. They have their own advanced back-up and replication methods to cope with this (as well as preemptively retiring old hardware). The actual disk you're accessing could fail as you download



Mint's Timeshift is an excellent and fuss-free way to carry out back ups, but for this outing we have a few more tricks up our voluminous sleeves.

a file, yet you might not even notice the delay as a back-up volume is transparently spun up.

## HOW ORGANISED IS YOUR SETUP?

“A significant obstacle to having a watertight back-up regime is how disorganised we are with our files...”

From an end user point of view, these services offer an invincible hold for their data. Invincible that is, until someone hacks your account or until you accidentally delete those files. Then, unless you had your own replication strategy, it's game over, man.

## » YOUR \$HOME AWAY FROM HOME

Off-site backups are an excellent idea, since they mitigate not only against hardware failure, but also against combustion and burglarisation. A10-drive raid array with triple redundancy will be duly decommissioned by fire or flood, or will be an attractive target for a thief with a penchant for storage.

Unfortunately, the cost of cloud storage (or renting/running your own off-site storage) as well as physical bandwidth means you probably don't want to store everything remotely.

Putting a monthly cost on your data is tricky. The likes of Google and Dropbox have free storage tiers, but you may not wish to support them. There are other offerings: Box (<https://box.com>) gives you 10GB for free (or unlimited for £12/month), but finding a solution that will integrate with your Linux desktop isn't straightforward. At the end of our feature we'll look at *Duplicati*, which can connect to all kinds of offerings.

If you trust a hosting service, then you can rent your own VPS (20GB offerings

are around \$5/month and the likes of Digital Ocean will even throw in daily backups for an extra \$1/month). This way you get your very own Linux box in the cloud, which you can do anything (*anything?—Ed*) you like with.

Just using it as a back-up target isn't the best value for money, but having a VPS affords one many options. Since this is data you probably want kept private, we can use a solution like *Duplicity* to encrypt everything before sending it cloudward.





# Fragile data

Data is precarious. We must protect it from all kinds of things, including fat fingers and cosmic rays.



**W**e believe it's useful to consider the scope of things we'd like our backups to enable us to recover from. Hardware failure is probably what comes to mind, but the scope of this is quite wide. An external hard drive is convenient for casual backups, but if it's subject to being moved around a lot then this will expedite its demise. If you back up to an another drive inside your PC, and you also use that PC for high-octane gaming or crypto mining (*does anyone still do this? – Ed*), then all that heat might end up cooking both copies of your precious data. So if possible we strongly recommend backing up to a separate machine.

We should consider not just the drives themselves failing, but also failure of the rest of the machine they're connected to. It's unlikely, but a PSU failure could take

harder sell, but they'll guarantee you the precious minutes to shut down your back-up machine cleanly in the event of a power outage.

## Backing up to optical media

Back in the day, it was popular to back up on to optical discs (CDs, DVDs, Blu-rays). These use a layer of dye on to which is 'engraved' by a laser inside the drive. CD-Rs and DVD-Rs use an organic dye, which over time becomes unstable. Since people have been using these media for the best part of three decades, there's anecdotal evidence that suggests this degradation can happen within a couple of years. Current science on this, assuming discs are stored ideally (out of direct sunlight, not subject to wild temperatures or scratching) estimates they might last anywhere between five years and a century. And for precious data it's not worth the risk. So your first step to healthy backups is to move any data you have stored on optical media to something that hopefully will stand better the test of time.

Easier said than done, it turns out. As we saw earlier, modern hard drives aren't infallible, and we'd caution against using an older drive for back-up purposes. There are all sorts of premium hard drive options, such as for NAS or home security purposes, but if you're not going to be writing tens of gigabytes a day you can get away with something more budget friendly. If you do plan on using your back-up drives to store anything else consider doing so with a separate partition, so at least if something goes wrong there, damage may be localised.

Apart from referring to the build-up of legacy code inside software, "bit rot" also refers to the phenomenon of files slowly corrupting over time. Even if your hard

## CHOOSE YOUR HARDWARE WISELY

"Modern hard drives aren't infallible, and we'd caution against using an older drive for back-up purposes."

out pretty much every component of your PC. As could a power surge. With modern filesystems and drives, data at rest (files that you aren't doing anything with) will be safe during a power outage, but if this happened while a backup was in progress then the results could be pretty devastating. Surge-protected power strips are probably worth the extra few coins, but they're not invincible. Uninterruptible power supplies (UPSes) are a

S.M.A.R.T. information								
Device information			Attributes		Self-test logs		Extended information	
ID	Attribute n...	Flags	Value	Wo...	Thr...	When failed	Raw value	Type
4	Start_Stop_...	-O--CK	100	100	0	-	3445	old-age
5	Reallocate...	PO--CK	100	100	50	-	2	prefail
7	Seek_Error...	PO-R--	100	100	50	-	0	prefail
8	Seek_Time...	P-S---	100	100	50	-	0	prefail
9	Power_On_...	-O--CK	79	79	0	-	8414	old-age

Keeping an eye on your drives' SMART statistics can warn of impending hardware failures.





This wallpaper is 20 years old. Witness the cruel effects of a few cosmic rays (centre) and then a few more (right).

drives and filesystems are in perfect working order. In our daily lives we're bombarded by (mostly harmless) cosmic rays that in an unlucky scenario might excite a tiny area of the storage platter (or NAND chip for SSDs) 'flipping' the values of the bit that was stored there. There are other causes of silent corruption, but gamma rays are the most interesting. You might wonder, in an era of petabytes and exabytes, what the effect of flipping a single bit makes. Allow us to show you. This Python code flips a random bit in a file called **test.jpg**:

```
import os, random
filename = "test.jpg"
f = open(filename, "r+b")
filesize = os.stat(filename).st_size
randombyte = random.randrange(filesize)
randombit = random.randrange(8)
print('Flipping bit', randombit, 'of byte', randombyte)
f.seek(randombyte)
byte = f.read(1)
f.seek(randombyte)
newbyte = (ord(byte) ^ (1 << randombit)).to_bytes(1, 'big')
print(byte.hex(), '->', newbyte.hex())
f.write(newbyte)
f.close()
```

Feel free to transcribe it and save it as **bitflip.py**. Then copy any image you like to the same directory and name it **test.jpg**. Just like real cosmic rays, this is destructive and so make sure you do have another copy of any valuable image you run this on. When you're sure, run the script ( `python bitflip.py` ) to simulate a quark of corruption hitting the **test.jpg** file you put in place. Now open the file (or just look at the screen shot if you can't be bothered) and see what's become of it. Maybe it's indiscernible from the original, maybe it has some slight corruption, or maybe it won't even open. Different filetypes will corrupt in different ways (there's nothing jpeg-, or even image-, specific about the code and cosmic rays don't discriminate), for example flipping a bit in an uncompressed bitmap image will very likely just change a single pixel. You can run the script multiple times to simulate multiple cosmic rays: it's a not very efficient way of making glitch art (for another interesting way to do this, using *Audacity*, see Alex Cox's tutorial in **LXF242**). In addition, feel free to show us a neater way to implement this in Python, or indeed any other language.

Silent corruption of files, or silent failure of drives (where they continue to be able to read and write, but occasionally do so inaccurately), is hard to mitigate against. Conventional RAID (see box, right) won't defend against it – for example, a two-drive RAID 1 array can tell you the two copies it has of a particular file are different,

but it won't know which one is corrupted (or indeed if both are). There are programs that can detect corruption by keeping track of hashes of files, but keeping large indices up to date and checking them is time-consuming. Over the page we'll show you how next-generation filesystems like Btrfs use checksums to provide self-healing capabilities. As well as all their other features such as RAID and snapshots.

It's worth mentioning that users are, on the whole, more error prone than hardware. Accidental deletions and misconfigurations are all part of the game. It's also worth considering the damage that user-installed ransomware might do. A nasty variety could scramble not just data on that user's machine, but data anywhere on the LAN that user had access to, including the back-up drives. Potentially even remote data could be affected too. Keeping your backups in cold (offline) storage is one way to protect against this threat. It may be worth the small inconvenience of having to plug everything in at back-up time.

## » MAKING THE CASE FOR RAID

One mitigation against drives failing is RAID (redundant array of independent/inexpensive disks). The idea here is to use a bunch of drives (usually, but not necessarily, the same size) to provide redundancy (or speed in the case of RAID 0, which we won't discuss). Common RAID schemes vary from simple mirroring across two or more drives (RAID 1), to more complicated arrangements that store parity data across at least three (RAID 5) or four (RAID 6) drives.

With RAID 5 for example, you can have a three-drive array such that if any one drive fails then data can be reconstructed from the other two drives. Unfortunately, for large drives recovery is a lengthy and I/O intensive process, and there's a possibility of another drive failing during the rebuild. You can experiment with the numbers at <https://magi.github.io/raid-failure>, but it's reasonable to caution against using, say three 4TB consumer drives in RAID 5 (which would give you a total capacity of 8TB, but also a 53 per cent chance of a unrecoverable error during rebuild). Instead, if integrity means more to you than capacity, use the drives in RAID 1 formation. You only get 4TB total space this way, but gives you double the redundancy.

You'll often hear it said (including by us) that RAID is not backup and you should heed this wisdom. However, RAID in tandem with backup is certainly a good thing, and so we'll look at setting that up later on in the feature.



Our trusty NAS uses RAID and is housed in a box very much like this.



# No-nonsense backups

Discover how you can go from back-up zero to data saviour in just a few clicks, with the excellent Déjà Dup.

**D**eciding what you want to back up can be tricky. However, it's worth considering taking a different approach for system files and your user data. In our Mint feature last month we looked at *Timeshift*, which is Mint's equivalent of Apple's *Timeshift* or Windows System Restore Points. *Timeshift* is well

supported in distros other than Mint and is an excellent way to mitigate against updates (or users!) breaking the system. Just roll back to a previous snapshot and block that update or don't do that configuration change. If your system files become damaged then it can help you restore good versions, and if you want it can take care of your personal files, too. Yet you should perhaps not do that, since you may end up inadvertently backing up much more than is useful (*Spotify* caches, say). Other programs will give you more granular control of this.

## Turn disaster into opportunity

You may also prefer to use operating system breakage as an opportunity to upgrade to a newer distro or re-installing the current version, in which case the *Timeshift* approach is no use. Installing Linux (on a good day) takes less than five minutes, and if you prepared beforehand, you can more or less replicate your system in around the same time. On Debian-based systems you can make a list of all currently installed packages with

```
$ dpkg --get-selections > installed_packages.txt
```

If you keep such a file updated (even on an ad hoc basis) and are able to recover it, then you can use it to



Using a disk usage analyser such as Filelight will help you identify any other large directories in your home folder that you needn't waste space and time with.

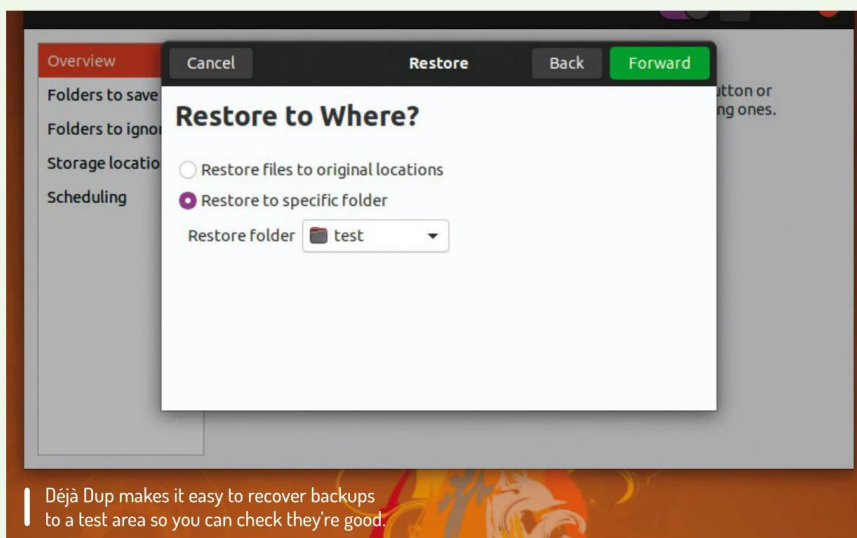
## » MAKE SURE YOU CAN RESTORE YOUR BACKUPS

Having backups is no good unless you can restore your files. If your system is unbootable, this can be problematic. With *Timeshift* or *Déjà Dup* you can just fire up a live disc, point the appropriate tool at your backups, and tell it where to put them.

It's worth testing your backups from time to time, by recovering some random files to a temporary location. Some data might need special treatment and it's worth thinking about this. This is particularly true if you're recovering additional data to a freshly

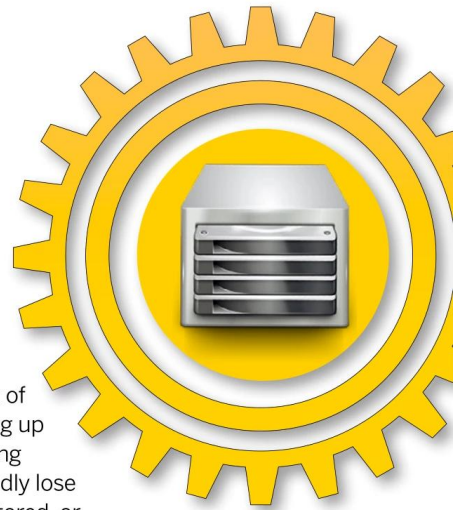
installed distribution, for example on a new machine after hardware failure. Files might contain data that's hardware specific, such as a reference to a particular partition ID in **/etc/fstab** (which tells Linux which filesystems to mount) or a particular network device. This can generally be resolved with some post-recovery tweaks, but it's worth being aware of before you overwrite configuration files.

For servers, extra care should be taken. Any services running on them, be it traditionally or through Docker or a VM, might involve a bit more subtlety than putting the original files back in the same places. Carefully consider how and where data for each application is stored. For example, getting your website working again might involve the Apache configuration, the CMS application and, most importantly, the data in an SQL database. Databases should be regularly exported to a portable format that can be included in regular backups.



Déjà Dup makes it easy to recover backups to a test area so you can check they're good.





reinstall those packages. That way we don't need to worry about backing up programs (the large `/usr` and `/lib` directories, for example). We should back up configuration files too, which live in `/etc` or `/var/lib`. If you know the services and daemons that you have configured yourself, then you can cherry-pick individual files from here. Obviously, this becomes more complicated for servers, but in a connected world we may as well make use of the fact that everything is in the cloud (dude). For this reason there's no critical reason to back up your Steam library, or indeed your towering collection of *Linux Format* DVDs (*ah, I remember them, they'll be back – Ed*).

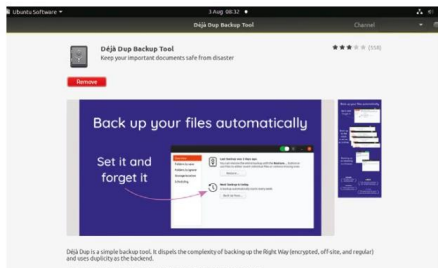
Whatever your data habits, there are at least a few personal files that you should back up. Password manager keyrings, browser bookmarks, work documents

and photos spring to mind. Your own digital proclivities and priorities will determine whether this amounts to megabytes or terabytes.

We're going to look at backing those up quickly and easily with *Déjà Dup*. The software can back up to a local or network drive, and can also back up to cloud services such as Google Drive. Best of all, it can set up a schedule so that backing up is carried out automatically. After all, having backups is all well and good, but they rapidly lose their usefulness if they can't be easily restored, or if they become corrupt.

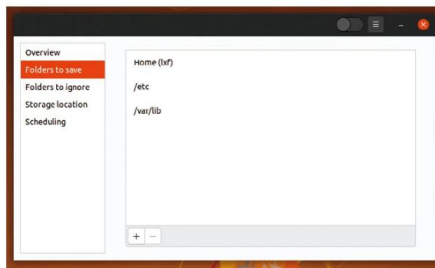
On the next page we'll look at a slightly more advanced approach to backups...

## EXPERIENCE DÉJÀ DUP



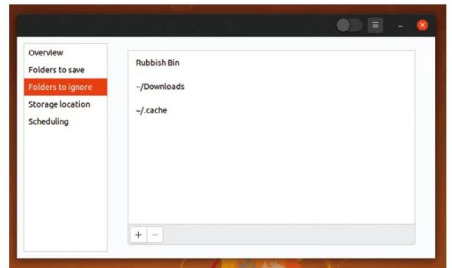
### 1 Install Déjà Dup

Déjà Dup is available as either a Snap or Flatpak and is almost certainly in your distribution's repositories. If you're using Ubuntu then it might even be installed already – it's named *Backup* in the Applications menu.



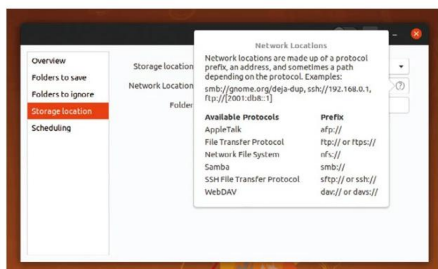
### 2 Choose folders to back up

You know the answer to this better than we, but to start with we'd recommend selecting the `/etc` and `/var/lib` folders to catch useful system configuration files as well as your `/home` directory. If you do work outside of your home directory or store photos, music and musings on a separate drive in `/media`, then include these, too.



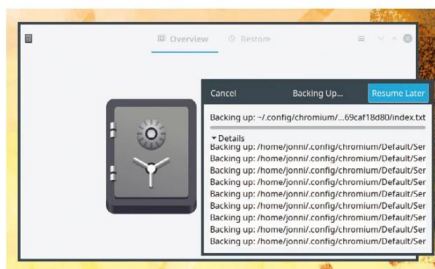
### 3 Choose what to exclude

Click the Folders to Ignore button. As discussed, you can safely omit the `~/cache` directory (press Ctrl-H to make hidden files visible), and possibly `~/Downloads`, too. You may also have large VMs or databases in the `/var/lib/libvirt` or `/var/lib/mysql` directories that you don't need to back up, so make sure these are excluded.



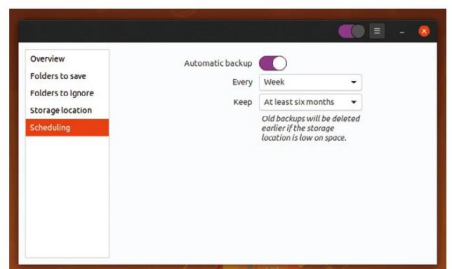
### 4 Choose a target

*Déjà Dup* can back up to a local drive (internal or external) or a remote share (for example, a server running SSH or Samba, or any services connected through Gnome or KDE). Choose Local Folder or Network Server as appropriate. Use your file manager or the `du` command to find out how much space is required.



### 5 Begin backing up

You'll be prompted to encrypt your backups with a password, which you probably should do. Then hit Forward and backups will commence. Once completed you can return to the program at any time and refresh the backups manually, which will be much quicker than the initial backup. Use the Restore tab to look at your backups.



### 6 Set a schedule

Hit the Automatic Backups switch to have the backup be refreshed every week. You can change this to daily if that better suits your requirements. Do keep an eye on how much space is being used though. *Déjà Dup* does things incrementally and so only changes are stored. Study the `.manifest` files to see what's changing.



# Backups in depth

Use snapshots and cloud backups to keep your data safe with minimal effort – just way we like it!



**H**ardcore terminal users may prefer to back up using the command line tool *Rsync*. This can be automated via cron (see p54) jobs (or systemd timers), and encryption can be added into the mix. *Rsync* is an efficient protocol (so only changed chunks within files get updated) and all it needs to work remotely is an SSH connection. It's a tad complex for the novice user, but do check out the manual page if you're interested.

A common gotcha is that it's easy to get one's slashes wrong when passing paths to *Rsync*, which during backup will be inconvenient and during recovery could turn one problem into two. Enter the popular command line tool *Duplicity*, which uses the *Rsync* library (but has a much nicer syntax) and can connect

(Copy on Write) snapshots feature to do things slightly more efficiently than the traditional *Rsync* method it uses on other filesystems. CoW means that when a file is changed, a new file is written out and only when that transaction has completed successfully is the old copy marked for deletion and metadata updated. This offers resilience, in the event of a power outage, say. It also enables incremental snapshots to be taken at the filesystem level, by keeping hold of the changes between old and new files. At first these seem counter intuitive, since a snapshot initially takes up no space, but then grows as data changes.

If you don't use *TimeShift* (but do use Btrfs), you can still take advantage of Btrfs snapshots. In fact, they're especially useful for filling in the gaps, say in a weekly *Déjà Dup* back-up schedule. For example, you could, keep half a dozen hourly snapshots, and then single daily snapshots going back seven days. As long as there's enough space, this enables you to swiftly roll back your files to that state they were in this morning, or last Tuesday (*hey, that's when this feature should've been in – Ed*). OpenSUSE uses Btrfs by default these days, and its YaST package manager leverages this by taking snapshots before and after each update. A similar hook is available for *Apt* or Arch Linux's *Pacman*.

Btrfs uses its subvolume feature for snapshots, and you can also do fun things such as store snapshots from one Btrfs volume on a another, possibly remote Btrfs volume. Rather than getting our hands dirty with that though, we'll look at OpenSUSE's *Snapper* tool, which you can install on any distro. This will make easy work of creating and scheduling snapshots, which can also be used for system backups or backing up user data. This is only going to be useful for Btrfs filesystems, so if you're not using it consider doing so for your next install.

Install *Snapper* on Debian-based systems with

```
$ sudo apt install snapper
```

*Snapper* can only snapshot subvolumes, not run-of-the-mill directories. Since the root directory is a subvolume, there's no worries snapshotting that, but if you only want to give individual directories the *snapper* treatment, you'll need to convert them to subvolumes first, which is one for the manual. Note that snapshots don't include subvolumes that are part of the subvolume being snapped, so you'll need to create individual configurations for each subvolume. Having **/home** as a separate subvolume then becomes handy, since in a disaster it and the rest of the system can be recovered independently as required. There's also a case for making subvolumes of other noisy locations such as **/var/cache** and **/var/tmp**.

By default, *Snapper* will use *systemd* (it can integrate

## MAKE USE OF ALL YOUR OPTIONS

“Modern filesystems such as ZFS and Btrfs provide features that can assist or augment your backing-up schemes.”

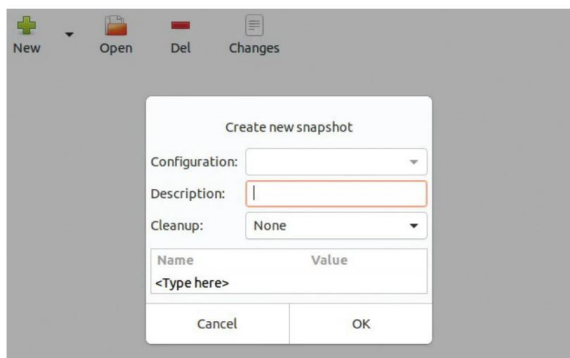
to all kinds of third-party storage (including Amazon S3, Dropbox, Google Drive, Microsoft OneDrive) as well as sending your backups over SSH or locally.

*Déjà Dup* is a frontend for *Duplicity*, which you might have noticed by the filenames in your newly populated back-up directory. If your *Déjà Dup* backups are stored in the directory **/mnt/backup**, then the command

```
$ duplicity collection-status file:///mnt/backup
```

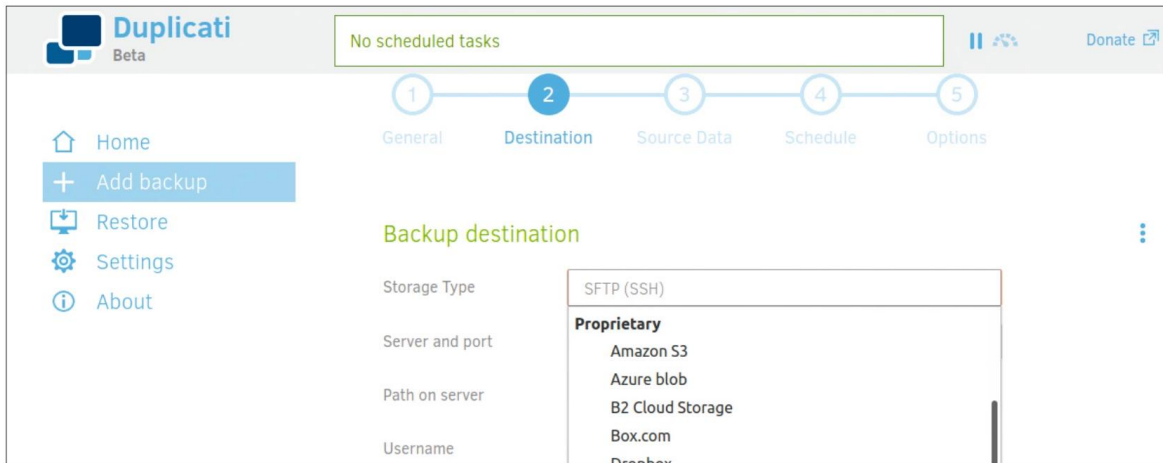
will show you some information about them. *Duplicity* has all kinds of features which *Déjà Dup* doesn't make use of, so if your back-up requirements are a little more involved then it's well worth a deeper look at this.

Modern filesystems such as ZFS and Btrfs provide features that can assist or augment your backing-up schemes. We didn't talk about it, but if you store your *TimeShift* backups on a Btrfs volume it can use its CoW



A Python-powered GUI for Snapper is available at <https://github.com/ricardovieira/snapper-gui>.





Duplicati can back up to more cloud services than can comfortably fit in this screen shot.

with crons too) to schedule 10 hourly, 10 daily, 10 monthly and 10 yearly snapshots, which depending on your purposes might be overkill (especially if you do this for the root subvolume). To tame this a little edit the config file you created in `/etc/snapper/configs/`. For our six hourly and seven daily suggestion earlier, add the following lines:

```
TIMELINE_LIMIT_HOURLY="6"
```

```
TIMELINE_LIMIT_DAILY="7"
```

Then set the longer periods to "0". The timeline service will start on reboot, but if you want to enable it immediately do:

```
$ systemctl start snapper-timeline
```

## Duplicati illuminat

To finish, we'll look at one final back-up tool, which is a little more advanced. *Duplicati* is a client-server affair with a web-based interface. You can use it to centrally manage backups for all the machines on your network (including Windows boxes), but we'll just cover running it locally.

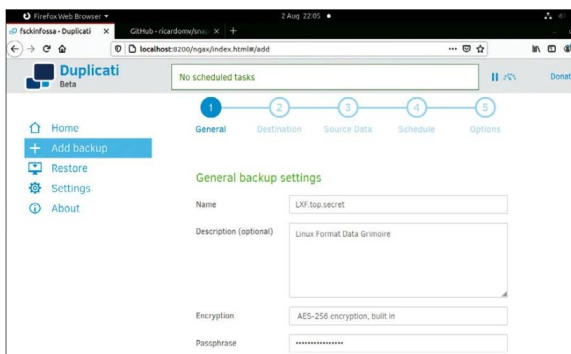
*Duplicati* is available in DEB and RPM packages from <https://duplicati.com>. The 2.0 version is technically still a beta, but it's well-enough regarded that we're talking about it here. It's a Mono application that depends on an awful lot of packages and so it's easiest to install it via Gdebi:

```
$ sudo apt install gdebi-core
```

```
$ sudo gdebi duplicati_2.0.5.1-1_all.deb
```

Start the service with `systemctl start duplicati` then visit the web console at `http://localhost:8200`. The wizard will walk you through setting up your first backup

*Duplicati* supports backing up to lots of proprietary



Thanks to our Duplicati server running in a lead-sealed box, team LXF can sleep soundly knowing their data is safe.

cloud services, but can also back up locally, use SSH, or connect to your Nextcloud instance via the WebDAV protocol. Once you've chosen the destination, choose what to back up, either using the tree view or by adding paths directly. One gotcha here is that the Home folder in the User data drop-down refers to that of the user running the service, rather than the one doing the backups. So choose your home folder explicitly in the Computer drop-down. If you're that way inclined you can filter files, either explicitly or using regular expressions. Now hit Go, watch the progress bars and enjoy that warm fuzzy feeling that goes with knowing your data is safe.

Finally, don't forget about your phone. If you don't want to rely on Google or Apple, you can use the *Syncthing* app to synchronise your phone with your computer, then integrate that into your regular backups. Next, choose a schedule, by selecting which days it should run, and how long to keep backups (the Smart retention option is a good one if you're not sure). Then hit Save and you're done. **LXF**

## » JOINING THE DOTFILES

Hardcore Linux users are rightfully proud of the myriad configuration files they've painstakingly tweaked to suite their needs. These so-called dotfiles (because they are named `.vimrc`, `.Xresources`, `.i3config` etc) live in your home directory, most prevalently in the FDO-mandated location `$HOME/.config`, and store settings for just about every application you use. Search Github for dotfiles, and you'll see some power users devote an entire repo to their most prized specimens. This not only enables (in the spirit of free software) others to borrow or adapt these files (so that their Vim and Zsh can look cool as well), but also serves as a rudimentary back-up facility.

Even if you've never hand-edited a dotfile they can still be handy thing to back up, otherwise you'll have to redo any application-level changes from default if you ever end up re-installing or restoring your OS. You should be careful though – some of these files contain personal information, including account credentials. So one should carefully vet what one puts on a public website, or indeed any cloud service. There's also limited value in backing up everything in `.config/` since lots of chaff accumulates in here. Furthermore, if you restore older versions of configuration files they may not work with newer versions of their respective programs. Read more about the joys of version controlled dotfiles at <https://opensource.com/article/19/3/move-your-dotfiles-version-control>.



A man with dark hair and a slight smile, wearing a grey hoodie over a black t-shirt, stands with his arms crossed. He is positioned in front of a large, stylized background of binary code (0s and 1s) on a light brown, textured surface. The text 'PRESSURE VESSEL' is overlaid in large white letters across his chest.

# PRESSURE VESSEL

Jonni Bidwell gets on the video chat to talk video games and message buses with D-Bus guru **Simon McVittie**



**S**imon McVittie is a Debian developer and a consultant for Cambridge/Canada-based open source-ers Collabora. Last year he was awarded a UK Open Source Award for outstanding contributions to FOSS.

Simon's in charge of D-Bus which is, among other things, how all the different bits of your desktop talk to one another. He's also been working with Valve Software on an exciting project named Pressure Vessel that aims to have Steam leverage the magic of containers, to make for a smoother cross-distro experience. And to alleviate the Steam runtime's current dependency on libraries from eight years ago.

Locked-down Jonni Bidwell took a break from the endless trips between the computer and the snack cupboard (that for nearly four months have been his life), to talk to Simon about interprocess communication mechanisms, gaming and *Wine*. And also to yabber on about his ancient Eee PC that, despite conventional reasoning, to this day remains alive and well. There's an allegory about print media here, but let's not go there and instead listen to Simon's wisdom and erudition...

**Linux Format:** I'm told you won a prestigious award at the UK Open Source awards. Bravo. Your colleague Mark Filion told me you were shy about this, but that this recognition is well-deserved. Can you tell me a little bit about the awards?

**Simon McVittie:** The awards are to recognise companies that are doing good in the open source space – particularly for major projects like *LibreOffice* – and individual developers. In the case of the latter, people who have established themselves for a while, such as myself, and also students and upcoming developers.

**LXF:** You're a Debian contributor, and have been for a while. Can you tell us how that journey started?

**SM:** I started using Linux back at sixth form, so aged 16 or 17. I started using Debian not long after I got to university. I can't remember when exactly I started contributing to it – 2002 or 2003 probably. My first job out of university was for a company that was using a lot of Linux stuff for their product, without actually doing any open source themselves. So they were taking a Linux stack and building



proprietary software on top of it. While I was there I met one of the founders of Collabora, who jumped ship to found his own consultancy. A year or so later I was getting fed up with that job, so I decided to join him.

**LXF:** Did you study computer science?

**SM:** No, Mathematics. I describe myself as a recovering mathematician.

are some really exciting developments connected to algebra and cryptography, but they're far too complicated for me.

You're the main developer for the D-Bus subsystem, which is something that casual Linux users may not have heard of, but which nonetheless holds lots of bits of their desktops together. Let's pretend I've done my homework and am not just reading from your

## SIMON'S GENEROUS VIEW OF D-BUS

"It's not the fastest, both in terms of latency and throughput, it's not the most expressive and it's not even the easiest. But it's among all those factors that it's a decent compromise..."

**LXF:** Me too. High five. One day I hope to be free of the group theory nightmares.

**SM:** The way of thinking is very useful for programming: being able to think through all the possible results of something, and making sure all the cases are enumerated rather than just saying "it'll probably be fine". The actual mathematics I've probably only used once since getting my degree. That was to audit someone's implementation of Shamir's (the S in RSA cryptography) secret sharing, and to ensure it had the security properties that it claimed.

**LXF:** Mercifully, I haven't had cause to use group theory for many years now. There

company's website. "This essential piece of Linux infrastructure connects everything from chat clients to the Nautilus file manager, the Common Unix Printing System, and music players". That seems like no mean feat.

**SM:** The way I like to think of it is as the interprocess communication (IPC) mechanism for when you don't have fancy requirements. It's not the fastest, both in terms of latency and throughput, it's not the most expressive and it's not even the easiest. But it's among all those factors that it's a decent compromise, absents any elaborate requirements. That's kind of not a bad reflection of what I do in software. I've become the generalist that





has become mistaken for a specialist, by people who know less about things than I do. So when people come to me and say, "I've been trying to do this obscure thing with D-Bus" or "why is it not faster?" or "why does it not have this obscure feature?" or that kind of thing, my answer is often "Are you sure you want to be using D-Bus?". Because it's general purpose and does a lot of things, people want to assume it does all the things, and does them fast. But the price of being a jack of all trades is being master of none. And D-Bus is very much that.

**fun with ConsoleKit and Udisks. I'm glad I don't have to worry about that stuff anymore, to be honest.**

**SM:** I'm always slightly ambivalent about lightweight desktop environments. On the one hand lightweight software, minimum requirements, yeah great, why would you not want this?

Then you look at a full fat desktop like Gnome or KDE, and everything in there is there for a reason. And most of those reasons are pretty good engineering. Obviously some of it is historical inertia that we wish we didn't have. But the vast

writing applications you can pretty much rely on all the desktop environments having this stuff. And if you're writing a desktop environment then you really better have this stuff otherwise it'll be considered deficient. And having D-Bus as a sort of lowest common denominator IPC is a big part of that.

We have **.desktop** files, which were originally KDE's answer to Windows 95's shortcuts, but over time have expanded into something we can do a lot of stuff with. They've more or less turned into an application registry effectively. Just before this call I was answering a mailing list thread about how to figure out which terminal emulator to use to launch TUI applications, which are **.desktop** files. Which is a really hard problem.

**LXF:** There are all kinds of subtleties when you start thinking about this stuff, or even try to make an Eee PC friendly desktop. At first glance it can all seem unnecessarily complicated, and we get reader letters decrying this fact. Sometimes they even draw comparisons to the relative simplicity of Windows, which we don't think is fair.

**SM:** If we had a single monolithic Linux platform, then the answer to most of these things would be, "We have a monolithic stack that does this well, here's how we're going to use it, job done." So that would be easy if we only had Gnome, or only had KDE. And if we only had both of those things, then it's harder but still do-able. But because we have MATE, Cinnamon, LXDE, LXQt and anyone else who has stapled together a desktop environment out of duck tape and string, making all of those interoperate is quite a challenge, because there's so many moving parts. You can't do a flag day API transition across all these desktop environments.

Gnome is probably the closest to being managed like the sort of monolithic project where you would be able to do a flag-day transition, but even Gnome isn't completely unified, and people have their little territories. But you can't do a flag day across all of the **freedesktop.org** world. And that means there's a danger of things ossifying, you know, situations where we can't change X to Y because desktop Z relies on X and no one's going to change it.

**LXF:** Can you explain a bit more about freedesktop.org (FDO). It seems like a thankless task trying to give guidelines for desktop environments that are all so very different.

## SIMON'S FREEDESKTOP.ORG DREAM

"The sort of utopia of FDO is to have all the desktops implement one side of the protocol, and all the applications implement the other, and have the two pretty much interoperate."

**LXF:** I guess my first experience of it, and all these other desktop things we take for granted, was when I got my beloved Eee PC back in 2008. By necessity I got into lightweight desktops, and discovered all the luxuries you lose when you just run a barebones window manager, all the bits you have to wire together yourself.

D-Bus definitely figured in there somewhere, I guess for notifications maybe, and I also remember all kinds of

majority of it, if you don't have it already, then you're going to end up reinventing it in a lightweight setting. And if you don't understand some of what's missing then you're probably going to reinvent it badly.

A lot of the **freedesktop.org** stuff around things like D-Bus, but not exclusively D-Bus, is providing a level playing field for all the different desktops. Having a baseline for doing certain things in a desktop-agnostic way. And if you're

## » LEISURE TIME

**LXF:** What have you been playing lately?

**Simon McVittie:** My recent gaming has been replaying *Saints Row 3*, which is native Linux, yay! And also trying to get back into *Shadow Tactics*, which is also native Linux. It's really good – I recommend it if you're at all a fan of the stealthy sneaking assassinating genre. I've just got to the last level, which is ridiculously hard it appears. I now see why I bounced off it a few months ago. I've heard great things about *Desperados 3*, which is by the same people (Mimimi Productions) and is very much a spiritual successor—*Shadow Tactics* itself was inspired by the original *Desperados* games by Spellbound Entertainment. And Mimimi have been an excellent steward of that. Oh and there's the itch.io *Black Lives*

*Matter* bundle too. That seemed to involve all of indie gaming.

**LXF:** Thank you for reminding me about that. I should probably not buy games for a while.

**SM:** The curse of the neverending library. I do tend to use 'has Linux version' as a filtering mechanism. Supporting developers who are making games for the sake of making games, rather than just making the most money. I've been impressed with the Linux ports of the *Tomb Raider* reboot and the *X* series by Egosoft. Linux is treated as a first-class citizen there. But also I'm surprised with how well Proton is working now. Even on games that were buggy on Windows at release. (cough cough, *Batman Arkham* series) it works amazingly well



**SM:** Some of those desktops like Gnome and KDE are quite closely integrated, so they're customisable, but the biggest decision point is that you choose to use it or not use it. At the other end of the scale, some desktops are more "here's a box of bits for you to make your own desktop"; and most desktops are somewhere in between.

**Freedesktop.org** (FDO) started off as the X-Desktop Group, and I think you still see XDG in the specifications, at the beginning of environment variables. I think its origins were in the protocols window managers used to communicate with applications. The core X protocol is really, really basic and you needed things like hints to put on a window to say, for example, "please don't allow this window to be maximised" or "please don't draw the titlebar, I'm doing that". There's no point in having a Gnome hint, a KDE hint and LXDE hint, and whatever else, and having your poor application author set all of them. Because that doesn't scale.

So the goal of FDO was, at least for the subset application authors would want to address, to provide a baseline standard for these type of things. So the sort of utopia of FDO is to have all the desktops implement one side of the protocol, and all the applications implement the other, and have the two pretty much interoperate. Over time FDO has absorbed things like .desktop files, which I think originated in KDE 3. D-Bus is a part of FDO, that started as a reinvention of, or at least took a lot of inspiration from Gnome's use of CORBA and KDE's DCOP. FDO also specifies XDG base directories for putting your configuration and your cache files in. So that's useful because you only have to specify one cache directory to exclude when backing up, instead of about 30.

**LXF:** Yes, we talked about that in our back-up feature. The `.cache/` directory does get pretty big pretty fast. But it would be much harder to write about if every application and every desktop stored these things in different places. But I still imagine it's tricky trying to decide how to add new features, or change old ones in a truly agnostic way.

**SM:** At its best, developers of two or three of the big desktop environments say, "There's a problem, we need some sort of a solution, here are our requirements, here's something that looks like a good enough answer, let's do it." In that case everyone has consensus and agrees and it's implemented. At its worst, one of the

desktop environments says, "Here's a problem we need to solve, here's what we think is a solution" and there's no response and the poor developers are left shouting into a void.

The big weakness of things like FDO is that they're not anyone's main job. Even for people that do a lot of volunteer development, they're not even anyone's main volunteer work. And if anyone tries to take these things on as their main volunteer work, then I think people from the desktop environments would tend to be suspicious of that. Because these are things that have to come from the overworked people that do things like Glib,

GTK, Qt, the core desktop libraries. They have to come from people working at the infrastructure level, because there's no point having a half-baked idea from someone that only does applications, and then having the desktop compositor authors or the library authors look at it and say, "Well this isn't actually implementable. This requires us to be psychic."

**LXF:** You gave a talk at FOSDEM, probably one of the last times something like civilisation was able to freely gather together. Well, I suppose we had Cheltenham race day just up the road in March, too...





**SM:** Yes, I think back in February it hadn't sunk in how bad things were going to get.

**LXF:** Yes it's been a wild ride. In that talk (which readers can check out at [https://fosdem.org/2020/schedule/event/containers\\_steam](https://fosdem.org/2020/schedule/event/containers_steam)) you covered some of the interesting work you've been doing for Valve. They have a

**difficult job ensuring Steam titles run on all distros. Their solution has been to provide their own baseline, the so-called Steam Runtime, which as far as I can tell is a bunch of libraries from Ubuntu 12.04. But your talk points to this changing?**

**SM:** The stuff from my talk is very much the next generation. The current generation of Steam titles still rely on its runtime, which is like the pragmatic version of the LSB. Instead of having some mythical cross-distribution thing, they just picked a distro which at the time they did it was popular and well-supported, namely the then-current version of Ubuntu. They told developers to target that runtime, and then they could guarantee their games worked with Steam.

When those titles run they do so in a container that makes the user's distro look like the one being targeted.

This has its limitations and we're bumping up against that now. Game developers quite reasonably want to use library stacks that aren't eight years old. They want to be able to use the new features of the gnutls or whatever libraries.

Moving the baseline from Ubuntu 12 to anything newer is a really hard sell, because for every game you fix by bumping the baseline version, you're going to break one. And some of the ABIs that were available in Ubuntu 12 no longer exist – they've been superseded by something similar but not the same. So you can't just bump up all the versions and use Ubuntu 14.04 or 16.04 or whatever and call it a day. It's particularly hairy around Glibc, for complex reasons around the runtime linker. I now know

far more stuff about this than I ever wanted to.

**LXF:** Sounds like cursed knowledge.

**SM:** Cursed and hard won knowledge. Although my colleague Vivek, who is hacking on runtime linkers for one of our ideas for the next generation of library compatibility, knows even more. It involves scary rebindings and relocations and things. That's for another Steam-related project called libcapsule, which interested readers should check out (an overview of the technology is available at <https://people.collabora.com/~vivek/dynamic-linking/segregated-dynamic-linking.pdf>). It's an interesting and fun project but doesn't really work yet. But we are using some of the supporting code for libcapsule in these attempts at newer Steam runtimes. We're actually using all of libcapsule except the library – that's the entire point of it, which is ironic. All the ancillary tools around it include lots of that hard-won knowledge about libraries, so we might as well reuse it.

One idea that has been batted around has been bumping up the versions of everything in the Steam runtime, and then rebuilding everything, but all against the old version of Glibc. And that's a massive amount of work. The amount of bootstrapping and building the new thing against the old thing, and the newer thing against the new thing that distros have to do. We'd have to replay that across an eight-year boundary. And we don't want to do that. It wouldn't even give us particularly good results, because we have to think about long-term sustainability. It's not enough that we have the runtime, we also have to be able to maintain the runtime.

So we don't want to run around patching every library to have different behaviour, because then we have to merge those patches into new versions forever, and we'd really rather not do that. And that's where the containers come in. They're an attempt at solving some of the runtime issues. There are some design documents at <https://github.com/ValveSoftware/steam-runtime> that I refer to in my FOSDEM talk if people want to find out more about what's currently true and what we'd like to be currently true, runtime-wise.

**LXF:** Your talk discusses your Pressure Vessel work, which aims to modernise this runtime. Can you talk through that a bit please?





**SM:** The idea with Pressure Vessel is that a lot of the contortions the current Steam runtime has to go through are because it's just a giant `LD_LIBRARY_PATH`. So you bundle a set of libraries from Ubuntu and mangle the search path so that they're found before the libraries from the host operating system. Your game uses those libraries and everyone's happy. Except that they're not, because that completely breaks open source graphics drivers.

The Nvidia driver gets away with it by heroic efforts on the part of Nvidia, because they compile their driver on Red Hat from 2010, I don't know – something really ancient anyway. That way it works on everything. The open source drivers quite reasonably tend to assume... well, for example I built Mesa recently on Debian Testing, so it would be reasonable for those drivers to assume they'd have at least the library versions from Debian Testing. So when presented with all these older libraries bundled with the Steam runtime it breaks. Mesa has also started using C++ for shader compilers through LLVM, which makes the ABI problem a whole lot harder.

That was how the Steam runtime originally worked and that approach was quickly abandoned. How it now works is that it compares libraries from the host system and the Steam runtime, and tries to use the newer ones. And that works, except when it doesn't. Because comparing library versions is far harder than anyone might have reasonably expected. Recently I've been giving Pressure Vessel library-specific knowledge, so it knows to use comparison X for library Y. So with Pressure Vessel instead of making this giant search path, you make a container, a lot like Flatpak or Docker or whatever. In fact I recycled an awful lot of LGPL code from Flatpak to make this. Inside that container is the only library stack we want, so the game has no choice but to choose those libraries.

Part of the idea is that a game developer can do their QA against Pressure Vessel, which is a really quite strict system. If it works on that then it's much more likely to work everywhere. Whereas if they did their QA on the Steam runtime with, say, the latest Ubuntu LTS, will it work on Arch Linux? Who can say? Will it work on older Ubuntu? It might, but probably won't though. Having a giant test matrix of all the distributions isn't really feasible. So now, of course, we have a giant test matrix for Pressure Vessel on all

the distributions, but at least we only have to do that once.

**LXF:** I saw recently that there's actually been kernel work going on related to gaming. Something about syscall user redirection, which I think was something to do with Wine.

**SM:** Ah yes, my colleague Gabriel [Krisman Bertazi] is leading that. There's a good write-up about it on Linux Weekly News (<https://lwn.net/Articles/824380>). The idea there is that some Windows games have started making direct system calls to the kernel. The normal way to do things, on both Linux and Windows, is to use some high-level API, either in Glibc or the Windows equivalent, that will do the actual talking to the kernel for you. Or on Linux you can go a step down, and tell Glibc to make a syscall on your behalf, and it'll go and do the actual trapping into the kernel. So that way you can do shiny new syscalls that, even if your Glibc doesn't know about them, it can still handle them generically. Which means if



## ON BRINGING STEAM UP TO DATE

“Game developers quite reasonably want to use library stacks that aren't eight years old. They want to be able to use the new features of the gnutls or whatever libraries.”

you want to trap all syscalls on the way through, then you only need to catch them in Glibc.

However, Windows game developers have now started making syscalls directly into the Windows kernel. And libraries have no way to intercept that, without being monstrously slow, and unfortunately Wine is a bunch of libraries. So if you want to run your game in Wine, or on Steam through Proton, it works fine until it hits one of these syscalls. So maybe it's being told “do syscall 42 with these arguments”. On Windows syscall 42 might be open a file, but on Linux it might be reset the clock. So the kernel would go “this doesn't look like a clock” and your game crashes.

Gabriel's idea is to trap all the syscalls, except those that are made in a special way, and have the kernel redirect them into user space, so that Wine can deal with them. We have to bypass this mechanism for the syscalls that are actually made by

Wine, or the Linux libraries it depends on, because otherwise it'll be monstrously slow. In fact, you have to do a syscall to return from a syscall, so you'd end up trapped in some infinite recursion of failure otherwise.

So you have to have an exception, and have that exception be really fast. Because these are the syscalls Wine makes all the time. There have been a couple of attempts at this with completely different designs. It seems to be really well received anyway, that we have this kernel facility which is pretty much just for Wine. I'm not sure if it's useful for non-Wine things at all. And it's pretty much only for games, too – other software doesn't have these syscall fastpaths.

**LXF:** So kernel developers do care about gaming. Whodathunkit? Simon, thanks so much for your time, and best of luck for the future. **LXF**





CREDIT: Bayu Rizaldhan Rayes

# LibreOffice 7

Neil Mohr discovers what's new in this latest release of the office suite that is one of the crown jewels of the open source world.

**T**hese very words have been written with **LibreOffice 7.0** and there's nothing groundbreaking or earth-shattering about that, which is absolutely something to celebrate. **LibreOffice** is yet another first-class example of a successful – at least in terms of function – open source project. It's feature-full, stable, supports open standards, backed by a vibrant community, underpinned by a ethical foundation and is successfully maintained and developed over time by a wide spread of developers.

Whatever your thoughts on *LibreOffice* itself, a full office suite is an essential component in any prime-time computing ecosystem and *LibreOffice* is just that. We hate to admit it, but *Linux Format* has been somewhat remiss of late for overlooking both the offline *LibreOffice* developments, but also the online cloud implementations that are based on it, such as Collabora's *CODE* or *NextCloud*.

So with the milestone release of *LibreOffice 7.0* we thought it was high time that we revisited this sprawling office suite and remind ourselves how important a project it is and its accomplishments. Of

course, you'd be foolish to think anything is perfect, and *LibreOffice* has its shortcomings. Certainly, there are questions over its long-term sustainability of the ecosystem, with moves afoot to try and shore this up. However, a sign of a well-run project is that the 7.0 release timeline has been hitting its targets, so by the time you read this the final release of 7.0 will be out and ready to download in Deb, RPM and Appliance (just make the file executable) formats alongside Windows and MacOS builds.

So grab your build, donate some money and take the suite for a spin...

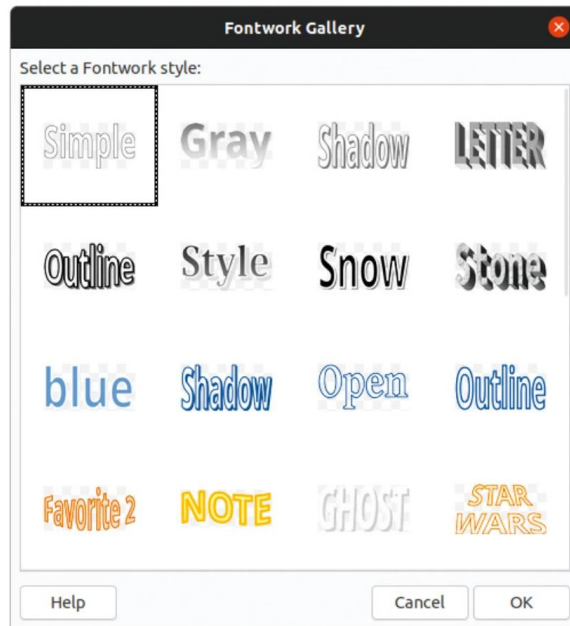


**A** key reason for the release of *LibreOffice 7.0* at this point is its support for the newly released Open Document Format v1.3. This was ratified by its governing body OASIS (<https://bit.ly/lxf267odf>) on 21 January 2020. Take a look at the box on the history of *LibreOffice* (below) and you'll see that from the initial days of its predecessor *OpenOffice*, supporting open standards and specifically the XML-based Open Document Format (ODF) has been a key tenet of *LibreOffice*. So it makes perfect sense that in an effort to promote and expand adoption of the ODF v1.3 format, *LibreOffice* ensures it's integrated into its next major release.

It's worth mentioning if you're unaware of the ODF it's now no small thing. In a similar way that copyleft software licences help expand the adoption of software, lengthen the life of projects and improve cooperation and interoperability, so the same benefits apply to a document. One objective of the ODF is to guarantee long-term access to data without legal or technical barriers. This alone has led many countries in the EU and across the world, including the UK as its civil service default format and NATO as an international body, to adopt open formats as a public policy issue.

On top of this over 600 companies promote the ODF through the Open Document Format Alliance. Corporations including Apple, Google, Adobe, IBM, Intel, Microsoft, the Wikimedia Foundation and Nokia, to mention just a few, work with the OASIS technical committee on developing and supporting the ODF. So as you can see, while "support ODF 1.3" might sound initially like an unremarkable addition there's a lot of fundamentals wrapped up in it.

The good news is that older versions of *LibreOffice* should have no issues opening new ODF 1.3 files even with the extended variants, though of course it's possible to select to export ODF 1.2 if required. The only issue you might experience is if you happen to be using another key new feature involving OpenPGP/GPG encryption. In this instance, only *LibreOffice 6.4.5* or newer will be able to open such documents.



All-new fancy type effects care of the FontWorks gallery.

As you can imagine, if you're dealing with large organisations, governments and military establishments then being able to validate documents might come in handy. First introduced into *LibreOffice 6* as an "experimental" feature, with the official introduction of digital signatures in the v1.3 release of the ODF, this is now an officially released feature. Digital signatures for documents and OpenPGP-based encryption of XML documents have seen improvements with change tracking, and additional details in the description of elements in first pages, text, numbers and charts.

Of course, the ODF isn't the only document format in the universe and *LibreOffice* can't survive in a vacuum because it needs to interoperate with other office and document formats. So there's good news for those who still need to support *Microsoft Office*, with the DOCX format support extending to 2013, 2016 and 2019 formats rather than just 2007 compatibility mode.

## » DECADES OF LIBREOFFICE!

The year of our Lord 2020 is an auspicious one for *LibreOffice*, and not just because of its 7.0 release. The open source code base will be 20-years old, while The Document Foundation that oversees the project turns 10 this year.

*LibreOffice* has its open source roots in *OpenOffice*, which first came out back in 2000. This was released by Sun Microsystems, which spun it out of its then-proprietary office suite *StarOffice*. This in itself dates back to the bad-old DOS days and *StarWriter*, which was developed throughout the late 1980s.

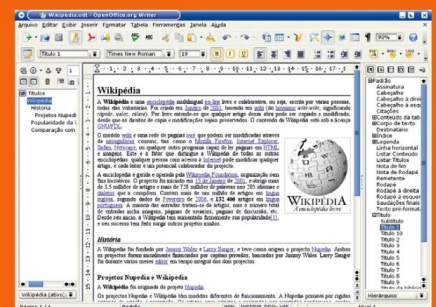
But *OpenOffice* was a hugely important release and from this the XML Open Document Format was developed and supported by *OpenOffice*. It's hard to qualify, but it's estimated

that between 15 and 20 per cent of all users had *OpenOffice* installed and the site registered 300 million downloads.

Despite Sun Microsystems promising a more open *OpenOffice* project run by a neutral foundation, the lack of movement worried many external developers. Michael Meeks had maintained a patch set of *OpenOffice* to make builds on Linux easier and accept external contributions. As time went on the *Go-OpenOffice* build started to offer more features than the upstream "official" *OpenOffice*.

So when Oracle bought Sun Microsystems and took ownership of *OpenOffice*, concerns about the stewardship of *OpenOffice* forced the situation. The Document Foundation

was formed in 2010 as a non-profit, independent, democratic foundation to develop an open source office suite, forking *OpenOffice*, building from *Go-OpenOffice* and creating *LibreOffice*. And that, as they say, was that!



Open projects tend to outlive their founders, and The Document Foundation is a good example.



LibreOffice's changelog states, "This mode is intended to ensure users of different versions of *Microsoft Office* can continue working together and documents created with older versions of *Office* won't look any different when they're opened in future versions of *Office*." So this mainly benefits *Word* users, where documents can make use of more features and *Word*'s bugfixes because DOCX 1.0 can be applied. Apparently this means *Word 2010* users lose out, but as TDF points out Microsoft has done the same since the *Office 2013* release, and so *Word 2010* users should upgrade to *LibreOffice*.

We'd say these are the big-title changes and are at the heart of the 7.0 release. If you flick through the changelog, you might get the (false) impression that not much has changed. Obviously, there's constant bug fixes, but many one-line features hide big background changes. We'll skip through some of the more interesting additions in particular order and fire up a list of some others.

### Semi-transparent text

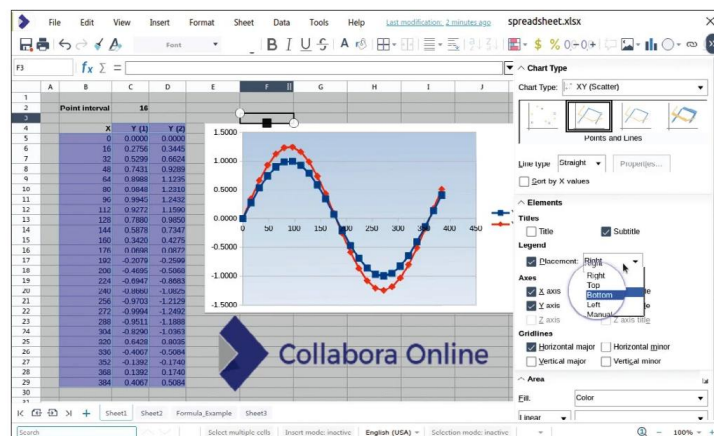
Introduced in two separate areas of the suite – *Writer* and *Draw/Impress* – creating semi-transparent text is now possible. This will enable you to add a bit of style to your presentations and documents.

## » LIBREOFFICE ONLINE

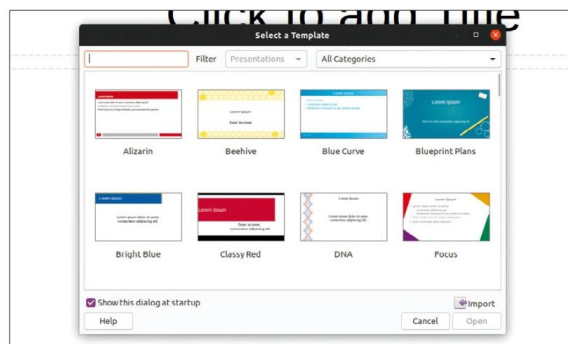
As we all know (or suspected), all the cool kids are online with their heads in the clouds! There is indeed an online version of *LibreOffice* and it's helpfully called *LibreOffice Online*. It offers a feature-full cloud-based collaborative document editing experience and all that good stuff you'd expect! But offering an online service requires infrastructure, support, maintained servers, continuous development... all that expensive good stuff! Furthermore, The Document Foundation has been deliberately structured so it's unable to offer that sort of activity directly.

So while TDF supports the developers and maintains the source code, it relies on third-parties such as Collabora with *Collabora Online* ([www.collaboraoffice.com/collabora-online](http://www.collaboraoffice.com/collabora-online)) and NextCloud ([www.nextcloud.com](http://www.nextcloud.com)) to offer packed solutions that work off the peg.

You can see the various release options for *LibreOffice Online* ([www.libreoffice.org/download/libreoffice-online](http://www.libreoffice.org/download/libreoffice-online)) these includes options from Collabora and NextCloud alongside the source and Docker container deployments.



Yeah, online editing, brew your own or spin up an offering from someone else.



LibreOffice's Templates have been updated to all support 16:9 ratios and some include theme support for colour matching.

Alongside the existing transparency effects for shape fill colours and shape border colours in *Draw/Impress*, it's now possible to add transparency to text within a text box. This can be applied as a paragraph or character-level style. There's also a Set Transparency percentage value that will dictate how transparent, if at all, text should be. The primary focus was *Draw* in this case, but this also helps PPTX support, because the importer/exporter now handles this for *Impress* documents, too.

Previously a feature in *Write*, transparency had been added to shape fills and shape borders. There are also plans to extend this to text, based on the work done with *Draw*. The upshot is that this also improves compatibility with *Word*, which supports the same feature in its DOCX format. You can access the new option in the Format>Character dialog.

### Glow and soft edges

It's a good example of the collaboration that goes on behind the scenes with *LibreOffice*, in how glows effects were implemented with the help of Collabora – that does an awful lot of commits – and SUSE. Two features that were implemented by developers Tamás Bunth and Mike Kaganski were object glow and soft-edged objects. This helps improve compatibility with *Microsoft Office* that offers the effect. The glow effect is available on shapes in all *LibreOffice* modules, but also on pictures with *Impress* and *Draw*.

Another new but related image effect is soft edges to images. Again, this is something that's offered in *Microsoft Office* but not replicated in *LibreOffice Impress*. It's actually a pretty useful feature, and makes it possible to add an alpha halo to the outside of an image, helping to blend it into the background.

### Artistic updates

A number of *LibreOffice* areas have been reworked to enable users to more easily create better-looking documents. The first is the Fontwork, which is similar to the WordArt gallery in *Microsoft Office*. The Fontwork gallery has been overhauled with fresh artwork for you to choose from. Find this new feature under the Insert>Fontwork menu.

*LibreOffice* maintains a gallery of default "objects" that you can drop into documents. These are scalable, can be edited and their number has expanded over the years. For the 7.0 release this collection was rationalised, ensuring that the available options are editable by the user and of a high quality. All previous collections are









**Cerys Lock** is the organiser of the Stafford Raspberry Jam and a general Pi enthusiast.

## » KEEPING ALL CHANNELS OPEN

Hi there, my name's Cerys and I run Stafford Raspberry Jam. It's a community-led event that enables people of all ages to meet, make and create with the Raspberry Pi. This helps to put computing into the hands of people and makes it possible for them to create amazing things with the help of the community. Usually, we meet once a month.

Unfortunately, due to COVID-19, and like many other meet-ups, we've had to stop meeting regularly at Stafford Library. This has meant that we've had to find different ways to communicate with one another. Some meet-ups have opted to go virtual with software such as Zoom or Microsoft Teams, but I didn't feel like everyone would want to log on at one specific time – it could be the children's bedtime for example (I understand this is what happens when people do actually go out to Raspberry Jams, but you're physically there which makes it much easier to interact with people). I also felt like one or two people couldn't get into talking about a project as only one person could speak at a time online.

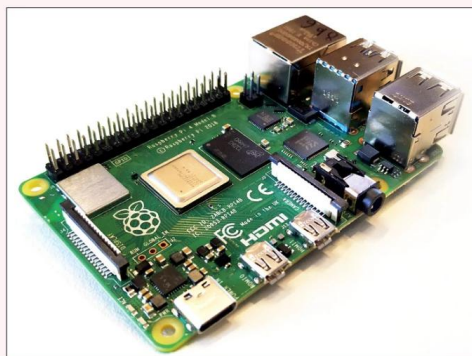
Luckily, I sent out an email to all the attendees. Within a few minutes, I had a reply from Phillip from OutServeWeb talking about the possibility of setting up a new website. After conferring, we decided to progress with the creation of the site – and you can see the result at [www.staffordraspberrypijam.co.uk](http://www.staffordraspberrypijam.co.uk).

If you'd be interested in writing a blog on the new website please use the Contact Us form at the bottom of the website. We hope you enjoy it!

# Self-hosting Raspberry Pi cluster of Pis

Could the humble Raspberry Pi stand up to Amazon and its Web Services business behemoth?

**W**e get the feeling that large sections of the population still see the Raspberry Pi as nothing more than a curio. For those people whose understanding of tech begins and ends with their smartphone that's understandable, but for years ISP Mythic Beasts ([www.mythic-beasts.com](http://www.mythic-beasts.com)) has been offering commercial web hosting and other online services, all run on clusters of Pis.



You might remember that the company bet Eben Upton a beer that it could run the Raspberry Pi website on the day the Pi 4 launched from a cluster of Pi 4s, which had to service tens of millions of visitors in a day.

Mythic Beasts already had experience running websites from Pi 3s, but had to retool Puppet for the updated version of Debian. It then moved a cluster of 18 Raspberry Pi 4s: 14 to handle dynamic web server needs (PHP/ Apache), two for static webserver needs and two cache servers. After testing, the cluster was moved to its data centre. After handling launch day and a month of smooth serving, the point was proved and the pint was served.

Read more on how the Pi 4 can serve web sites and take on Amazon Web Services here: <https://bit.ly/lxf267cluster>.

The Raspberry Pi 4 can happily and smoothly serve web pages.

## Radio PiPi

Feel the earth move.

**P**art of the Pi's appeal is the physical interaction necessary to make anything. We guess that goes double for projects that have an interactive element to them. So it's no wonder this interactive movable RadioGlobe has everyone talking! Move the globe and the Pi auto-tunes to a local station. Amazing! Learn more by visiting <https://bit.ly/lxf267radio>.



It's a novel way to tune in your radio.

CREDIT: Jude Pullen

## Pi in Space

Tasty winners!

**W**e've mentioned multiple times over the past year the various ongoing Astro Pi competitions. Now the winners to the Mission Space Lab 2019/20 the winners to the Mission Space Lab 2019/20 section have been announced. Their prize will include a live-streamed Q&A on 3 September with the ESA astronaut Colonel Luca Parmitano. You can find out the final projects here: <https://bit.ly/lxf267space>.



CREDIT: ESA



# 4tronix Picon Plus

**Les Pounder** needs to build a robot to help during social isolation. But can he build a robot butler to deliver more Pis to his table?

## IN BRIEF

An advanced motor controller for the Raspberry Pi that provides control over four motors, a single power source for the Raspberry Pi and the board. It enables control of up to four servo motors at once. Controlled over I2C and with a powerful Python 3 library, multiple boards can be stacked to control up to 16 motors independently.

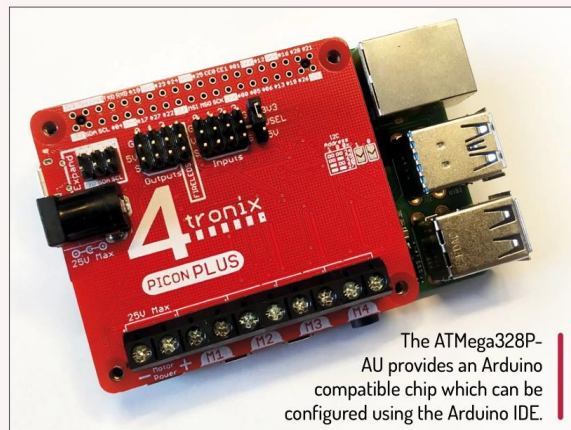
**A**dvanced robotics projects with the Raspberry Pi need a little more power than others. We need bigger motors, and more of them. We need servos to control steering and inputs to simplify sensors such as ultrasonic and IR. The 4tronix Picon Plus fills this void and offers a little extra for £30.

Designed for the HAT standard and compatible with all recent models of Raspberry Pi, Picon Plus occupies all 40 pins of the GPIO but it only uses four pins, I2C and power pins. This leaves the rest of the pins available for general use. To access the pins an extension header is required. More Picon Plus boards can be added on top of each other for a total of four boards and 16 motors! Each of the Picon Plus boards can be controlled independently thanks to the I2C interface and their unique identifiers.

Picon Plus provides four inputs and four outputs which offer SVG, Signal, Voltage and Ground for use with compatible devices such as servo motors. Inputs can be used with sensors or simple buttons depending on your needs. The device has screw terminals along one side of the board. Four of the screw terminal outputs are for use with up to four motors, all at once. This means we can build four-wheel drive robots, and because each motor can have their output adjusted using PWM (pulse width modulation) it gives us individual power control. Using four IFX9201 H-bridge motor controllers, each rated for up to 6A of current, we can use heavy-duty motors in our builds.

Power for the motors is provided via a DC barrel jack or via a screw terminal and we can supply up to 25V of DC power, which is regulated down to 5V, 3A to power the Raspberry Pi via the GPIO and to power our motors.

To control our build we have a simple Python 3 library, and a series of example files that demonstrate how to use the various functions for motor and servo control. It's great to see that 4tronix is now using Python 3 in its projects. Previously, it's only provided Python 2 libraries and examples and this was something which we have commented on in previous reviews. The Picon Plus library is designed for ease of use. The level of abstraction is



The ATmega328P-AU provides an Arduino compatible chip which can be configured using the Arduino IDE.

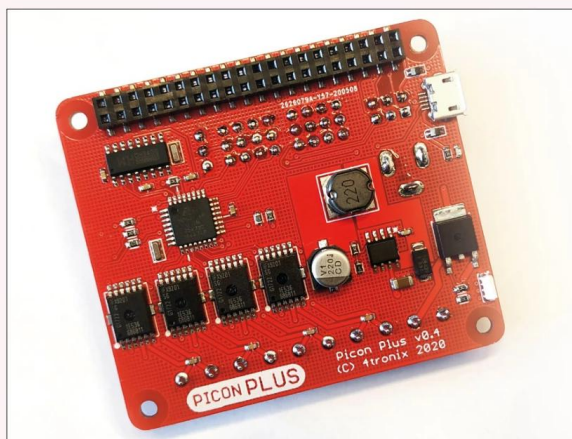
enough for new users to understand what a command may do, but also offers enough for advanced users to dig in a little further and tweak the library accordingly.

## Greater flexibility

What makes this board a little different to most motor controllers is an ATmega328P-AU that handles communication between the Python library and the motor controllers. The ATmega328P-AU can be tweaked using the Arduino IDE via a micro USB port. So if you need to make adjustments to how the board works with your project, you have the flexibility to alter the code and update the board using the Arduino IDE.

Picon Plus is simple to install, use and solves the problem of powering multiple high-current motors. The single power source and battery monitoring is superb and provides a tidy means to power the Raspberry Pi and the robot. It does everything so well and easily, for a very reasonable £30. The friction, the issues that we discover when building a robot are lessened with good hardware and software, and here we have both!

If you need a motor controller and are already well versed with Python and robotics, then Picon Plus is the ideal board with which to build a new project. **LXF**



Fitting on top of the Raspberry Pi GPIO, Picon Plus is a neat and refined board that offers maximum features in a small package.

## VERDICT

**DEVELOPER:** 4tronix

**WEB:** <https://shop.4tronix.co.uk>

**PRICE:** £30

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

A simple, no-nonsense motor controller that's easy to use, extremely powerful and configurable.

» **Rating 9/10**



## EXPLORER HAT PRO

# Assemble basic touch and motor controls

Get tinkering with **Les Pounder** as he demonstrates two projects using a board which may be old, but is by no means obsolete!



**OUR EXPERT**

**Les Pounder** is associate editor at Tom's Hardware and a freelance creative technologist. He blogs about his discoveries at [bigl.es](http://bigl.es)

**F**or this month's tutorial we go back to a board which is now five-years old, but still has plenty of life left in it. The Explorer HAT Pro is an electronics experiment powerhouse that comes with connections for touch inputs, dual motor drivers and analog inputs for analog electronics such as potentiometers. There are also protected inputs and outputs that are 5V Arduino compatible. Oh, and there are some LEDs and breadboard fitted on top just for good measure. We have two projects to build. Project 1 is a gentle introduction to using the board with capacitive touch inputs. Project 2 is a tank controller for two motors using analog thumbsticks.

With the Raspberry Pi powered off, set up the Explorer HAT Pro so that it connects to all of the GPIO pins and sits flush atop the Raspberry Pi. If you have any standoffs, these can be used to screw the Explorer HAT Pro to the Pi.

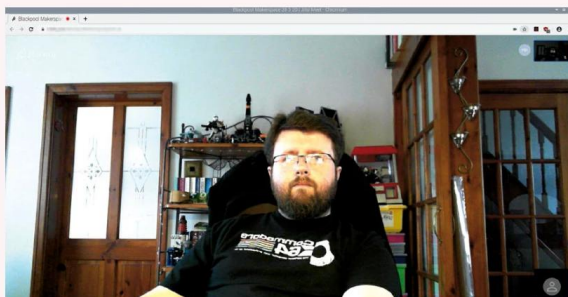
Connect your keyboard, mouse, webcam and so on to the Pi and then power up. Open a terminal and type the following to install the Python3 library:

```
sudo apt-get install python3-explorerhat
```

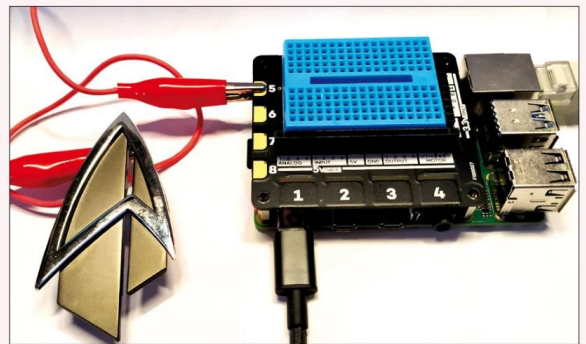
When complete, close the Terminal and open the Thonny Python editor, which is found in the main menu under Programming.

### Project 1: The power of touch

Capacitive touch surrounds us. Our phones, tablets and some laptops have a screen that we can touch and interact with. But did you know that fruit, aluminium foil and certain objects can be used as capacitive touch inputs? The Explorer HAT Pro has eight touch interfaces,



By touching the communicator we initiate a call on Jitsi Meet, directly in the browser. Make it so!



Using a crocodile clip and a Starfleet issue communicator, we can hail our colleagues on Jitsi Meet.

numbered 1 to 8. But 5 to 8 are designed for use with crocodile clips. In this short project we shall turn a metal object, in our case a Star Trek Picard communicator, into a touch interface that will start a video call with my local Linux User Group.

The first three lines of Python code import modules of prewritten code for use in the project. First we import the **Explorerhat** module renaming it to **eh** for ease of use. Then we import the **webbrowser** module, used to open web pages in the default web browser. Finally, we import the sleep function from the time module, this is used to add a pause in the code.

```
import explorerhat as eh
```

```
import webbrowser
```

```
from time import sleep
```

Next we create a loop, which will run the code for the project.

```
while True:
```

Inside the loop we create a test: has the touch input connected to input 5 been touched? When it's pressed, this function will report "True" and this means that the code will progress to the next line:

```
if eh.touch.five.is_pressed() == True:
```

What happens when the input is pressed? First, we print a message to the Python Shell, called the REPL (Read, Evaluate, Print, Loop) which states that "Hailing Frequencies are open".

```
print("Hailing frequencies open")
```

To open the default browser directly to the address of our online meeting, we call the **webbrowser** module

### YOU NEED

- > Any model Pi 40 GPIO)
- > Raspberry Pi OS

- > Project 1: Crocodile clip  
Metallic object or conductive object  
USB webcam

- > Project 2: Two analog joysticks  
Two DC motors  
Ten female-to-male jumper wires



with the function to open a new tab. This will open a new browser if there isn't already one open, or add a new tab to an existing window. You can read more about online video calls in the boxout.

```
webbrowser.open_new_tab("URL for the Meeting")
```

The final line of code is a pause for half a second, used as a "debounce" to prevent opening multiple browser tabs.

```
sleep(0.5)
```

Save the code as **Project1.py** and when ready click Run to start the code. Now press the capacitive touch input and watch as the video call starts in your browser.

## Project 2: Tank controls

For our second project we learn how to use analog inputs to control the speed of two motors. For this project we need to wire up two analog joystick inputs, similar to those used in console controllers to the Explorer HAT Pro. The joysticks are really potentiometers that alter their voltage output based on how far we push the stick. Two motors are connected to Motor 1 and Motor 2 + and - connections. Please refer to the diagram for detailed instructions.

Again in the *Thonny* editor we start by importing two modules of code. The Explorer HAT module is renamed to **eh** for ease of use, and we again import the time function from the sleep module:

```
import explorerhat as eh
from time import sleep
```

We now start the main body of code. Inside of an exception handling statement we create a while True, which will run the code within:

```
try:
    while True:
```

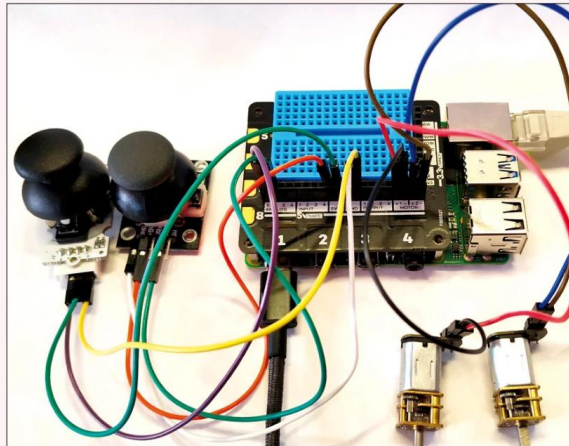
We create two objects, **motor1y** and **motor2y**, which will store a reading taken from analog 1 and 2, respectively. This reading is a voltage between 0V and 5V. The resting value, when the joysticks are centred, is around 2.5V. The returned value is precise and we don't need that level of precision so we wrap the output in a **round()** function and set the function to round the values to 1 decimal place. So 2.4443 becomes 2.4. We then print these values to the Python REPL for debug.

```
motor1y = round(eh.analog.one.read(),1)
motor2y = round(eh.analog.two.read(),1)
print(motor1y, motor2y)
```

Adding a pause of 0.1 seconds prevents the loop from demanding too much from the CPU. Two variables, **pulse1** and **pulse2**, are created and they use the returned value from the analog joysticks, multiplied by five to create float values that will later be used to control how long the Explorer HAT Pro LEDs will pulse for.

```
sleep(0.1)
pulse1 = motor1y*5
pulse2 = motor2y*5
```

To pulse the LEDs of the Explorer HAT Pro, we use the Pulse function. This will create a "breathing" light effect, similar to a laptop standby light. We need to pass four arguments to the Pulse function. Fade in, fade out, time on and time off are the four arguments that control the speed and look of the pulse. For fade in and out we will use the value saved into **pulse1**. For time on and off



Create tank controls for your next robot build using this quick and easy project.

we shall use **pulse2**. This will adapt the look of our pulsing lights as we use the joysticks:

```
eh.light.pulse(pulse1,pulse1,pulse2,pulse2)
```

To control the speed of each motor we use the joysticks. Pushing the sticks up or down the y axis will change the voltage output, which in turn is used to set the speed. The speed function requires values between -100 and 100 and our voltage output is between 0 and 5V, in reality it reaches 4.9V, so a little high school mathematics revealed that we need to multiply the voltage output by 20.4 to give us a figure between 0 and 100. This is then applied to both motors at once.

```
eh.motor.one.speed(motor1y * 20.4)
eh.motor.two.speed(motor2y * 20.4)
```

The final two lines of code handle when the user exits the code, by pressing Ctrl+C to create a "keyboard interrupt". When this happens the code prints "Exit" to the screen and then ends.

```
except KeyboardInterrupt:
    print("Exit")
```

Save the code as **Project2.py**, then click Run to start. The motors should come to life as the resting voltage is 2.5V, which equates to a speed of 51 for each motor. Try moving the sticks up and down to control the speed. Now all you need to do is build a tank! **LXF**

### QUICK TIP

The Explorer HAT Python module is simple to learn and offers many features. The outputs and LEDs for example, share the same commands and syntax. For help visit <https://github.com/pimoroni/explorer-hat/blob/master/documentation/Function-reference.md>.

## » FREE VIDEO CALLS

During the pandemic, one name has been on the lips of the world and headlines of technology sites, *Zoom*. But there are alternatives and one which we have been personally using for some time is called *Jitsi Meet* (<https://meet.jit.si>), a fully open source project.

Using *Jitsi Meet* is simple: we go to that link, create a new meeting and it gives us a URL that can be shared with our friends. That is all we need, no installation and no accounts. This URL was used in Project 1 to start a video call on our Raspberry Pi. This is all handled via a WebRTC JavaScript application that provides HD video calls for free! If you have an Android or iOS device then you will find official applications in your app stores.

But the simplicity with which we can use *Jitsi Meet* means that anyone can get on a call, and using Project 1 we can even set this up for push button control – handy for those who aren't comfortable with computers to stay in touch.

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



# ARCH

# How to run Arch Linux on the Pi

Jonni Bidwell installs Arch Linux in less than three days and discovers the best way to browse the web is from the console.



OUR  
EXPERT

Jonni Bidwell thinks he can hear the unmistakable sound of a 56K modem in the distance, calling him home

## YOU NEED

- > Raspberry Pi (any model)
- > 4GB or more SD card
- > Keyboard
- > Mouse
- > Monitor
- > Wired network

**T**he Raspberry Pi 4, thanks to its faster processor, USB connectivity and networking, is oft-touted as a desktop replacement. Be that as it may, we're going to show you how even an early model Pi can run a desktop... of sorts. We're going to use the community ARM port of Arch Linux to set up a lightweight desktop running *i3*. To add insult to injury, we'll combine the *w3m* browser, the *mpv* media player, and the popular *youtube-dl* script. This will enable you to browse YouTube (its many adverts and other streaming sites) from the comfort of a terminal window.

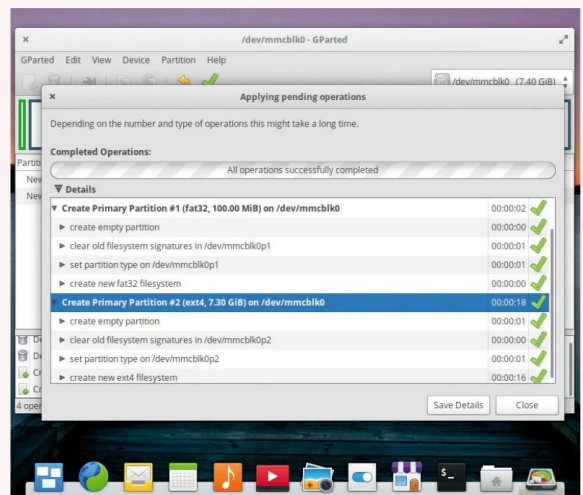
Arch Linux is generally recognised as being hard to install, and its ARM counterpart is certainly a far cry from installing Raspberry Pi Linux. However, once we've minted our SD card (which we'll soon see is a bit of a rigmarole), we pretty much have a working system. If you already have Arch installed on your PC, you can follow the second half of the tutorial to join in the console web-browsing fun.

## Prepare the SD card

Arch Linux ARM don't provide an OS image that we can write (using *Etcher* or the like) directly to an SD card, they instead provide a filesystem tarball that we have to extract manually. So the first step is to prepare our media. We'll assume you're working on a Linux PC with an SD adapter (with at least a 4GB card inserted) for this part. We'll also use *GParted* (a `sudo apt install gparted` away if you don't have it) to do the partitioning for us, but if you want to use *fdisk* at the terminal, follow the official instructions at <https://archlinuxarm.org/platforms/armv7/broadcom/raspberry-pi-2>.

Open *GParted* and make sure the SD card is selected in the top right (it would be a grim start to the tutorial if you were to inadvertently delete hard drive partitions here). Make sure there's nothing you care about on the card and then delete all the partitions.

Create a new partition using the button in the top-left. Leave it as a Primary Partition, make it 100MB and choose FAT32 for the filesystem. Then click Add. This will be our `/boot` partition. Create a primary data partition using the rest of the space, leaving the filesystem as ext4 this time. Hit the green check mark in the toolbar to apply the changes, heeding the subsequent warning.



Arch is all about doing things yourself, such as hand-crafting the partitions on an SD card.

Now open a terminal and mount our new partitions. SD partitions may be named as below, or as regular mass storage (for example, `/dev/sdb2`), or something else depending on your hardware. Use the `lsblk` command to make sure you get the right one:

```
$ sudo mkdir /mnt/piroot/
$ sudo mount /dev/mmcblk0p2 /mnt/piroot
$ sudo mkdir /mnt/piroot/boot
$ sudo mount /dev/mmcblk0p1 /mnt/piroot/boot
```

Now fetch the Arch Linux ARM archives. If you're running a Pi 3 or later, you may prefer to get the ARMv8 (64-bit) files instead in which case swap the 2 for a 3 (or 4) in the below:

```
$ wget http://os.archlinuxarm.org/os/ArchLinuxARM-rpi-2-latest.tar.gz
```

Extract the filesystem onto your newly created mountpoints. Don't be alarmed by the warnings about permissions (which the FAT32 partition can't handle).

```
$ sudo tar -xpf ArchLinuxARM-rpi-2-latest.tar.gz -C /mnt/piroot
$ sync
```

It'll take a couple of minutes to extract all that data. While it's doing that, dig up a monitor, keyboard and USB power for your Pi. The second command ensures all cached data is written before you remove the card.



We won't cover setting up wireless here (refer to the Arch wiki for that), but we will cover other basic setup via Ethernet. Put the card in and boot your new mini computer. Log in with username and password **root** then update the package keyring and database. For good measure, we'll update the system too:

```
# pacman-key --init
# pacman-key --populate
archlinuxarm
# pacman -Syu
```

It's good practice to set a system locale at this point, otherwise we might run into all kinds of weird bugs later. To do this, edit the file **/etc/locale.gen** and uncomment a line that corresponds to where you are. Proud UK residents should choose **en\_GB.UTF-8 UTF-8** for example. Then change **/etc/locale.conf** accordingly, so for the UK we'd say:

```
LANG=en_GB.UTF-8
```

## Launch i3

Let's jump straight in to getting **i3** up and running with:

```
# pacman -S i3
```

Then hit enter to install the whole **i3** group (which includes a screen locker and status writing scripts).

We'll need the X.org server as well, but we won't bother with a login manager. Instead, you'll be starting the GUI from the terminal, which requires the Xinit scripts. We'll also need a terminal emulator, and we're going to go with the lightweight, but Unicode capable, **Urxvt** (aka, **rxvt-unicode**).

```
# pacman -S xorg-xserver xorg-xinit rxvt-unicode
```

Now we need to create a simple **.xinitrc** in our home directory, so log out and log in as the standard user (username and password are both **alarm**, and you may want to create a new user later on).

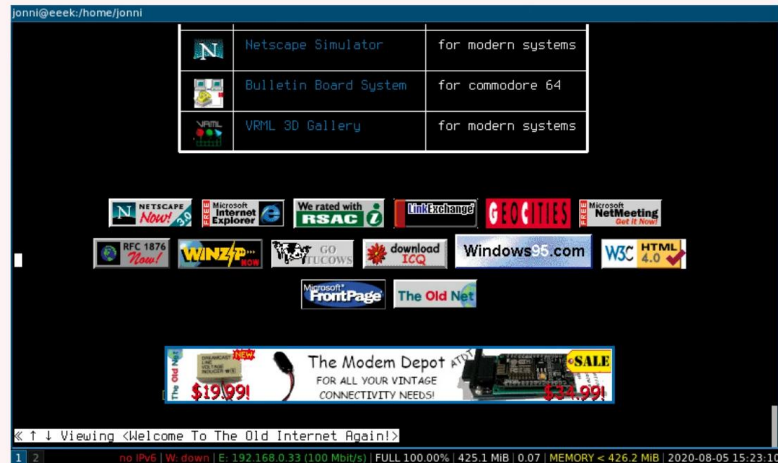
```
$ echo exec i3 > .xinitrc
```

Consider modifying your **.bashrc** if you want the GUI to start on login. Again, you'll find how to do this on this, as well as lots of helpful snippets for your **.xinitrc** file, in the Arch Wiki. You should now be able to start the display server with a simple **startx**.

Hopefully **i3** will start, in all its minimal glory. Use the Super (Windows) and Enter shortcut to start a terminal. It will be full screen (and slightly imposing) because that's how tiling window managers are. See the box (right) for details of how to configure fonts and colour schemes. For now, we'll get on with setting up glorious console web browsing. Switch to the root account with the **su** and then add the final pieces and return to the normal user account:

```
# pacman -S w3m mpv youtube-dl
# exit
```

Feel free to have a play with **w3m**, for example, **w3m https://linuxformat.com**. You can navigate with the cursors, jump between links with Tab and open them



Theoldnet.com reminds us of simpler times. W3M can sometimes load more advanced sites, but this varies with the weather.

with Enter. Check the manual for more. Anyway, **w3m** isn't quite ready for streaming video yet, so quit by pressing Q. **W3m** enables you to open URLs with different programs (so-called external browsers), so we'll set up the hotkey O to open files with the **mpv** media player. **Mpv** has bespoke support for **youtube-dl**, so after this quick change, we should be ready to go. Edit **w3m's** configuration file at **~/.w3m/config** and scroll down to the **extbrowser** line. Change it to read **extbrowser /usr/bin/mpv -ytdl** and save the file. Now press O on video links and they should open in **mpv**! The ARM package for **w3m** may not give you a config file by default, but you can just use that line on its own.

Oh, and because it would be terrible to end this tutorial without showing you how to get out of **i3**, Super-Shift-e is the combination you need! **LXF**

## QUICK TIP

Switch between the four default workspaces in **i3** by using Super-1 to Super-4.

## » CONFIGURATION STATIONS

There's an awful lot of customisation that these tools deserve and not a lot of space to talk about it. For example, if you don't like **i3's** default font (or if, as was the case for us, it doesn't render properly) you can install the Dejavu fonts with

```
# pacman -S ttf-dejavu
and uncommenting the
font pango:DejaVu Sans Mono 8
```

line, and commenting the previous (and slightly amorphous) monospace font declaration, in **~/.config/i3/config**. Check the documentation for any other changes you might like to make here.

**Urxvt** is configured by the **~/.Xresources** file. To change it to a hacker-style dark background and have a right-handed, simple scrollbar, use the directives:

```
Urxvt.background: black
Urxvt.foreground: white
Urxvt.scrollstyle: plain
Urxvt.scrollBar_right: true
```

To re-read this file without having to log back in, you can do:

```
$ xrdp .Xresources
```

Again, volumes have been written about beautifying your terminal through this file, and we'd encourage you to check those out.

» GET MORE JONNI EVERY MONTH! Subscribe now at <http://bit.ly/LinuxFormat>



# Scheduling tasks

Want to make time for hobby projects such as uncovering the name of Doctor Who? **Shashank Sharma** suggests automating recurring tasks.



**OUR EXPERT**

**Shashank Sharma** is a trial lawyer in Delhi. He collects geeky memorabilia, but wonders if sanitisers would qualify.

**T**he Occam's Razor principle is often paraphrased as 'the simplest answer is often correct'. Although this has long been thought as something that applies to science, this principle can also be used to determine the best software solution to a given problem. This is why tools like *Cron* and *At* remain so popular for automating recurring tasks. *Cron* is used for tasks you wish to perform repeatedly, such as installing updates, or performing backups. The *At* utility, on the other hand, can be used to run tasks only once, at the specified time.

But that's something that would only appeal to system administrators, we can hear some of you earnest readers complain! Although a critical skill for any administrator, automation of recurring tasks can be every bit as useful for desktop users. For example, perhaps you want your certain critical data to be backed up every day, or wish to delete the contents of the Downloads, or any other directory every week to ensure you have enough free disk space. You could perform each of these tasks manually, but the downside is that you might forget to do so every so often.

## Using Cron

An event that's scheduled to run at a specified time with *Cron* is called a *Cron* job. These events can be scheduled by editing the **crontab** file. The system wide file is located in the **/etc** directory. You can also create user specific *Cron* jobs, which are stored in the **/var/spool/cron** directory.

You can visit <https://crontab.guru> or <https://crontab-generator.org> if you need help mastering the different values used for defining cron jobs.

A **crontab** file may appear as no different than any other configuration file capable of being edited in any text editor, but you should only use the **crontab -e** command to edit the **crontab** file. This is because these files have specific formats, and running the **crontab -e** command ensures that all syntax errors in editing the **crontab** file are reported immediately.

The *Cron* daemon starts at bootup and runs a check every minute to see if any task is due. Each task in the **crontab** file is defined in a new line. Each line comprises six distinct fields:

**[Minute] [Hour] [Day-of-the-month] [Month] [Day-of-the-week] [command]**

As you can surmise, you can set up a *Cron* job to run specific commands, or even execute scripts. The first five fields are used to define the exact date and time when a task should be run:

### Defining a schedule

Field	Possible Value
Minute	0-59
Hour	0-23, where 0 implies 00:00am
Day-of-the-month	1-31
Month	1-12 or Jan-Dec (You can use first three letters to define the month)
Day-of-the-week	0-7, where 0 and 7 can be used for Sunday. (You can also use first three letters to define the day)

Additionally, the **crontab** file supports the use of different wildcards:

Wildcard	Purpose
*	Matches all values
,	Can be used to define multiple values for a field
-	Can be used to define a range of values for a field
/	Define intervals

For each entry in the **crontab**, each of the five defined conditions must be met for the scheduled command or task to be run. This can also lead to curious situations if you define a value for the [Day of the month] as well as the [Day of the week]. Consider the **30 0 1 \* 0** **command crontab** entry. Here, we've scheduled the command to run at 30 minutes past midnight on the first of every month, but only if the day also happens to

Complete the following form to generate a crontab line

Ctrl-click (or command-click on the Mac) to select multiple entries

**Minutes**

☒ Every Minute  
☐ Even Minutes  
☐ Odd Minutes  
☐ Every 5 Minutes  
☐ Every 15 Minutes  
☐ Every 30 Minutes

0  
1  
2  
3  
4  
5  
6  
7  
8  
9

**Hours**

☒ Every Hour  
☐ Even Hours  
☐ Odd Hours  
☐ Every 6 Hours  
☐ Every 12 Hours

Midnight  
1am  
2am  
3am  
4am  
5am  
6am  
7am  
8am  
9am

**Days**

☒ Every Day  
☐ Even Days  
☐ Odd Days  
☐ Every 5 Days  
☐ Every 10 Days  
☐ Every Half Month

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

**Months**

☒ Every Month  
☐ Even Months  
☐ Odd Months  
☐ Every 4 Months  
☐ Every Half Year

Jan  
Feb  
Mar  
Apr  
May  
Jun  
Jul  
Aug  
Sep  
Oct

**Weekday**

☒ Every Weekday  
☐ Monday-Friday  
☐ Weekend Days

Sun  
Mon  
Tue  
Wed  
Thu  
Fri  
Sat



be a Sunday. This event will thus not be executed until November 2020.

This is why it's far more common to define only one of these values, and use asterisks for the other. So, when an asterisk appears in the month field, the scheduled task will be run every month. Where an asterisk appears in the day field, the task will be run every day.

You can similarly use other wildcards to define a strict schedule for tasks to be run. The `30 0 * * 1-5 tar -cjf ~/projects-backup.tar.bz2 ~/Documents/projects` **crontab** entry can be used to schedule a back-up operation to be executed at 00.30 every day and every month, but the days are restricted to weekdays only.

The `30 5,17 * * 2,6 tar -cjf ~/projects-backup.tar.bz2 ~/Documents/projects` will perform a backup at 05:30am and 5:30pm every Tuesday and Saturday of every month. The `/` wildcard can similarly be used to define interval. The `0 14 */2 * [command]` will execute the specified command at 2PM every two months. The `0 */2 * * * [command]` entry can be used to schedule tasks you wish to execute every two hours.

You can have as many *Cron* entries as you want – just ensure that each entry is written in a separate line. When you're done, save the file and then exit the editor. Note that the entries in the **crontab** don't need to be arranged by date or time. You can add them in any order you want. The *Cron* daemon reads the entire file every minute to identify upcoming scheduled jobs and executes them.

For each entry in the **crontab**, you can also define multiple tasks or commands to be executed by using the `&&` operators. For instance, the `30 0 * * * command1 && command2` entry will ensure that both **command1** and **command2** are executed at 00.30am every day.

While the five fields are useful in strictly defining the schedule for *Cron*, you can alternatively use keywords to define specific time periods:

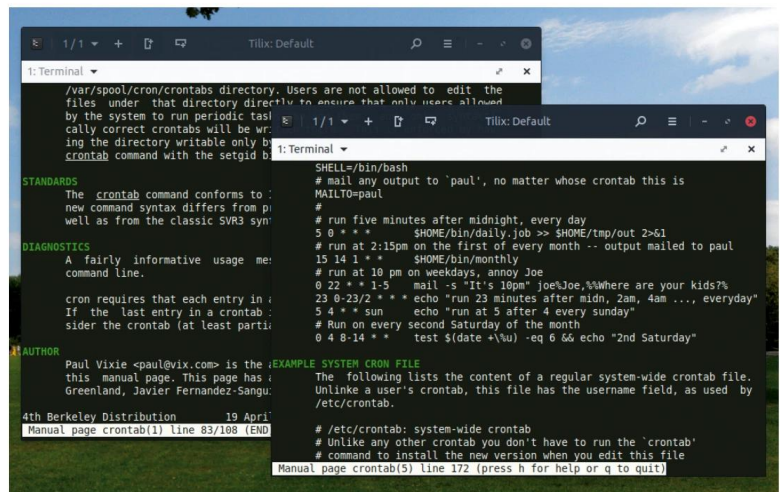
Keyword	Meaning
@reboot	Run once after reboot.
@yearly	Run once a year. Same as <code>0 0 1 1 *</code>
@annually	Run once a year. Same as <code>0 0 1 1 *</code>
@weekly	Run once a week. Same as <code>0 0 * * 0</code>
@daily	Run once a day. Same as <code>0 0 * * *</code>
@hourly	Run once an hour. Same as <code>0 * * * *</code>

The `crontab -l` command can be used to list all the scheduled jobs.

## Cron alternatives

The one obvious disadvantage of *Cron* is that it requires for the computer to be up and running. If your computer is switched off at a time when a scheduled job was to run, the task won't be executed until the next time when it matches the defined parameters. For instance, a task scheduled to run at midnight every Wednesday won't be run until the following Wednesday.

You can use *Anacron* for such situations, or for tasks for which you don't need to define precise times. Unlike



Run the `man crontab` and the `man 5 crontab` commands for instructions on how to properly create entries for managing scheduled tasks.

*Cron* which can be used to define the exact schedule for tasks, *Anacron* runs at bootup. So if a task is missed because your computer was switched off, it'll be executed at the next reboot. Although not installed by default, you'll find *Anacron* in the software repositories of most distributions.

*Incrn* or *Inotify cron* is another popular *Cron* clone. Instead of running tasks at specified times, it can be used to perform operations when certain changes are detected in the filesystem. For instance, tasks can be scheduled to run when a specified file is accessed or modified, a new directory is created, or a directory is deleted in a watched directory. **LXF**

## QUICK TIP

Days of the week are numbered 0-7 starting with Sunday, which can be either 0 or 7. If you think the week starts on Monday, use 1-7, otherwise use 0-6.

## » WORKING WITH AT

The *at* utility can be used to schedule tasks that need to be run just once. Scheduling jobs with *at* is easy because it accepts a variety of time formats. You can use keywords such as 'now + 1 hour', 'tomorrow', 'oct 30', '2pm tomorrow', and 'noon +3 days' to define the execution time for jobs.

When you run the `at <time>` command containing the date, you'll be dropped to the *at* prompt. Here you can enter the command or commands you wish to run. If you want to run multiple commands, press Enter after typing each. Press Ctrl+D when you're done. You'll be informed that a job has been created to be executed the entered commands, and *at* also confirms the time when said job would be executed:

```
~> at 5pm tomorrow
warning: commands will be executed using /bin/sh
at> mkdir ~/Desktop/tempdire-should-be-deleted
at> <EOT>
job 5 at Wed Jul 15 17:00:00 2020
~> atq
5      Wed Jul 15 17:00:00 2020 a linuxlala
4      Wed Oct 25 12:30:00 2020 a linuxlala
~>
```

The `atq` command can be used to print a list of scheduled jobs, and you can delete jobs with the `atrm <job number>` command.

» ENHANCE YOUR TERMINAL-FU Subscribe now at <http://bit.ly/LinuxFormat>



Credit: <https://snapcraft.io/symphytum>

## SYMPHYTUM

# Databases made quick and easy

**Nick Peers** reveals how to create your own simple databases in minutes with no coding or server experience required. It's open source magic!



**OUR EXPERT**

**Nick Peers** built his first database for his computer science GCSE back in the mists of 1988.

**T**hese days, databases are more routinely associated with powering websites and ecommerce systems. To the casual user they look impenetrable, involving connecting to third-party database servers such as SQL and hiding behind opaque languages like PHP. But at their heart, databases are simple tables of information: each row represents a single record, and its specific characteristics – such as name, colour, or whether it's currently in your possession or not – are recorded in columns known as fields.

If your needs are modest, then you don't need to learn any programming languages or tackle complex database software to put together a collection of information you can later search in various ways to find what you need from it. Indeed, many people build such

simple databases using spreadsheet software, but even here there's a whole new interface to learn and tools to find. There are only so many hours in the day.

Luckily, there's a whole host of simple, user-friendly tools that wrap up simple databases in software anyone can set up and use within minutes. Many are geared towards specific tasks, such as cataloguing a collection of plants, but there's one tool that offers all the features you'd expect of a simple database in a user-friendly format: *Symphytum*.

*Symphytum* is a throwback to the days when databases were simple to construct. Start by designing your database. First, define the fields that make up each record. Assign each field a data type, such as text, numbers, date, image, check box or drop-down menu, then design a form into which you enter the details of each record. The form also serves as a viewing tool for individual records, while a secondary table view provides you with an overview of your entire database, with the means to sort and filter the view using a simple keyword-based search tool.

There's no programming language to learn or no confusing interface to navigate, making *Symphytum* the perfect tool both for those with straightforward requirements as well as those looking for a gentle entry point into the world of databases in general.

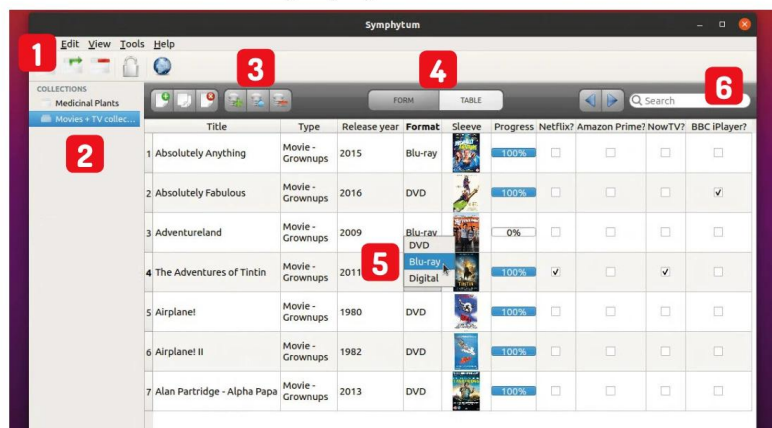
## Get started

You can install *Symphytum* a number of ways. Snap is the most convenient for later versions of Ubuntu, but you can also download your choice of .deb, .aur or .AppImage package from the program's website (<https://github.com/giowck/symphytum>).

Once installed, launch *Symphytum* (from the App Launcher in Ubuntu) where you'll see a small dummy database is already in place to help familiarise you with the program. There are two views: Form and Table. As we mentioned, Form is for primarily for data entry and reviewing individual records, while Table enables you to view all the records at once. You can also edit individual records from here, too.

*Symphytum* stores all your collections in a single file – **data.db** – that's buried deep inside a hidden folder. This is powered by an embedded SQLite database engine, so it's a standard format easily ported to other

## Master Symphytum's Tables view



### 1 Database tools

Use these buttons to create, duplicate or delete a collection, as well as lock your database form and set up cloud sync.

### 2 Collections

Individual databases are stored as a separate collection, but all your collections reside in a single .db file.

### 3 Field/Record controls

Use these buttons to populate your database with new (or duplicated) fields and records. Also delete unwanted items.

### 4 Choose your view

*Symphytum* provides just two straightforward views. Choose either Form for entering data and viewing single records, or Table for a wider overview.

### 5 Edit records

You can edit individual records in Table view as well as Form view – just double-click a field to do so.

### 6 Search

Quickly filter the records shown in Table view by typing a keyword into the Search box.



programs (more on that later). If you'd rather move the file somewhere more accessible then choose Tools> Preferences and click Browse... If you're considering backing up the database or storing it in a cloud folder for backup and sync purposes, check out the box (below right) for details.

Building a database from scratch is a simple affair with *Symphytum*: the step-by-step guide over the page reveals the basic process you need to follow in terms of setting up fields and designing the form for both data entry and reviewing individual records. And that's the program's beauty – you can have a fully formed database up and running in just minutes.

## Field types

As you've seen, each record is made up of information stored in one or more fields. *Symphytum* currently supports 12 different field types: Text and Number are self-explanatory: simple fields containing free-form text or numbers.

There are three date-related field types. Date enables you to record a date (with option of time) in one of three formats: DD/MM/YYYY HH:MM, Day Mon HH:MM YYYY, or YYYY-MM-DD HH:MM. You can opt to set the current date and time as the default value, and have the program alert you when the date is reached (useful for attaching a deadline or reminder to a specific record). The other two date field types – Creation Date and Modification Date – are automatically generated, and the best choices for when you want to know when a particular record was either created or last modified.

The Checkbox field is used for simple yes/no questions – tick the box to indicate yes, leave it unchecked to indicate no. Combobox enables you to define the record's data from a drop-down list. Unfortunately, there's no way to rearrange the list of options, so you might want to sketch these on paper first to ensure they're input in a logical order (such as alphabetically or chronologically).

A Progress bar makes it possible to define the progress of task-related records – in the example TV/films database we're using, you might want to indicate how far through a TV series you've watched by season. Once defined, you get to choose how many steps are displayed. Then, when creating or modifying the record, simply use the step controls (+/-) and the progress indicator bar next to it will update accordingly.

The Image field type enables you to attach an image file to the record, which is displayed as a thumbnail. All the usual suspects – from PNG and JPG to GIF and BMP – are supported. *Symphytum* stores a copy of these files inside the hidden files folder inside the same directory as your master **data.db** file, so we'd recommend favouring compressed formats like JPG where possible to keep the overall size down.

You can also attach files directly to records – click this and the original file is opened in its parent application, enabling you to modify it. Again, a copy of the file is placed in the files folder, so the original file you linked to will remain untouched. Finally, both Web Link and Email fields are also provided, which means you can create clickable links should you want them.

There's enough scope here to satisfy most people's needs, although we're a little disappointed there isn't (yet) a checkbox or combobox option that makes it



possible for you to set multiple values for a field. Perhaps a few email requests to the developers will persuade them to add this functionality?

## Input records

Once your form has been set up, adding a new record is as simple as filling in the form. Your record is automatically saved and updated, and then it's a simple case of pressing Ctrl+N or clicking the New Record button to move on to the next one. Any fields you've marked as required are clearly marked in red and you'll need to fill these in for the record to be accepted.

The Form view remains in design mode, so as you fill in records, you're free to redesign the form to better suit your needs as well as add new fields if you feel you've missed anything. You'll have to modify each record to fill in the missing details, however. If you're happy with your form design, click the lock button on the toolbar to lock the form design so you don't inadvertently make mistakes. Need to edit the form later? Simple: just click the lock button again.

Form view is perfect for viewing individual records, but as your collection grows, take the time to switch to

When entering records in Form view, click the lock button to focus on data entry and not form redesign.

### QUICK TIP

If the database scrolls slowly in Table view, choose Tools > Preferences > Advanced. Tick the Caching Images option first and if that doesn't help, choose Hiding Images to remove all images from Table view.

## » PROTECT YOUR DATABASES

There are two ways to protect your *Symphytum* databases: through manual backups stored elsewhere on your computer (or over the network), or via the Cloud synchronization button. Choose File> Backups to make (and later restore) a manual backup – just follow the wizard. Your databases and any file or image attachments will be bundled up into a single file with a .syb extension.

If you want either a real-time backup or the ability to work on your database across multiple machines or collaborate with others (*Symphytum* supports Windows and macOS as well as Linux), then a better option is to use your cloud provider to keep files in sync. Rather than move your database folder to a folder you're already syncing to the cloud, click *Symphytum*'s Cloud synchronization button.

For now, *Symphytum* offers three options: there's direct support for Dropbox and MEGA (choose the appropriate option then follow the prompts to connect your account), plus a handy 'Generic provider (folder based)' option. This latter option enables you to save your database in a folder being synced to the cloud using a native Ubuntu tool, such as Nextcloud or Google Drive.

If you do the link through cloud synchronisation, then when you open the database on your PC, it's made 'read-only' for other users who try to open the synced database on their own computers. They can view the file, but can't make changes to prevent sync conflicts.





## QUICK TIP

If you set up a Date field with the 'Show alert when date is reached' reminder in place, you can get a heads-up on impending due dates by selecting View>Date reminder list.

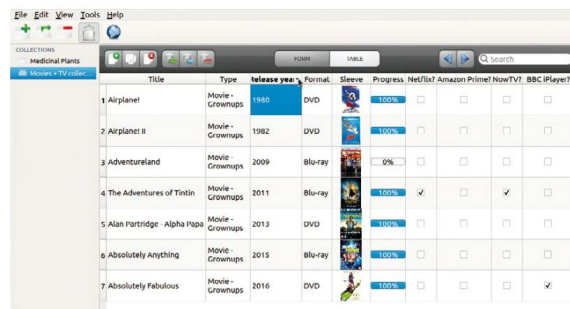
Kexi is one of the most-user friendly ways to step up from Symphytum, but it's still a jump.

Table View where you'll see a spreadsheet-like view of your database, with each record populating a single row and the fields listed in columns. Not only can you view and scroll through records, you can edit records too. Double-click inside a value to highlight it, then edit away.

Table View is also the place to go to rearrange the order of your fields – simply click and drag on the field title at the top of the table to move it to the desired position. If your screen gets a little cramped to display all your fields on-screen, click and drag on the drag handle to the right of a field to make it smaller (or larger) so they display their information better.

Too much hassle? Select Tools>Preferences>Appearance where you'll see options for customising both views. Click the Column width drop-down and change it from Adjustable to Automatic Width to have *Symphytum* make the choice for you (note that you won't be able to adjust the widths without reverting to Adjustable). You can also increase the size of each row by adjusting the line(s) setting.

If you want to tweak the look of your forms, then experiment with the Form View settings, with options for



The sort tool enables you to view your database in different ways – view movies chronologically rather than alphabetically, for example.

changing the background colour along with setting your own font size and style. There's also an option – Strategy for unused space – which you can adjust if you'd like *Symphytum* to spread your fields to fit the available space.

## Searching and sorting records

Browsing records can be done simply by scrolling through the records in Table view until you find the one that you're looking for. Select the record by clicking it, then switch to Form view. That might be fine for small collections, but like any good database, *Symphytum* offers two additional ways to track down the files you're looking for.

The Search box is the quickest way to filter records to show only those containing the keywords you type – as you start entering your search term(s) you'll see the list rapidly shrink. The downside of this approach is that it only works on fields containing letters or numbers – it ignores comboboxes and checkboxes, for example.

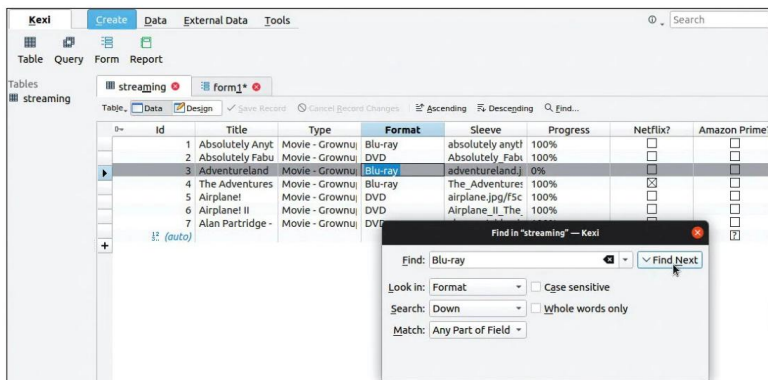
Another way to narrow searches is by sorting the table view by a different field to the first one you entered. Simply click a field title and you'll see it's highlighted with a small down arrow next to it, and the list will be sorted in descending order by that field. Click it again to reverse the sorting order or click another field title to sort by that instead.

## Share and export

You can print your entire database or individual records via the File>Print menu. Here you'll also find an option to generate a PDF. Don't expect magic results – the layout is the same whether printing a single record or all records (sadly, *Symphytum* doesn't yet enable you to choose a selection of records to print), which basically translates to each record being printed in series with each field on a separate row.

Another way to share your database is by syncing it via the cloud and sharing the folder in question with other *Symphytum* users – see the box on the previous page for details. Incidentally, you can override the automatic option whereby a database is 'locked' by the first user who opens it to prevent sync conflicts – select Tools>Cloud>Read-only mode to do so.

You can also share your database by exporting one or all records (again, no middle ground) as a CSV file via the File>Export menu. Select your encoding format and choose a separator (comma or semi-colon) before clicking Export. This should then be readable in most other databases, including *LibreOffice Base* (see the box, left, on relational databases). **LXF**



## >> WOT, NO RELATIONAL DATABASES?

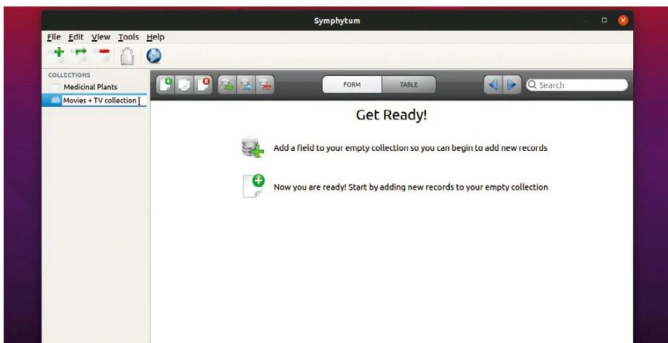
As of now, *Symphytum* only handles simple, so-called flat databases. Relational databases are separate databases that share a common field – known as a key value – that links them together. Taking our film and TV database as an example, if *Symphytum* supported relational databases, then we could use it to set up one database to store details of the movie itself, before linking it to another database that handles which streaming services the content is currently showing on.

If you find yourself outgrowing *Symphytum*, then one obvious step up is *LibreOffice Base*. It's quite a jump, however, which is why you might want to take a look at *Kexi* ([www.kexi-project.org](http://www.kexi-project.org)) instead. It's available through the *Ubuntu Software* store or as a Flatpak. Once installed, create a new, blank database stored as a file.

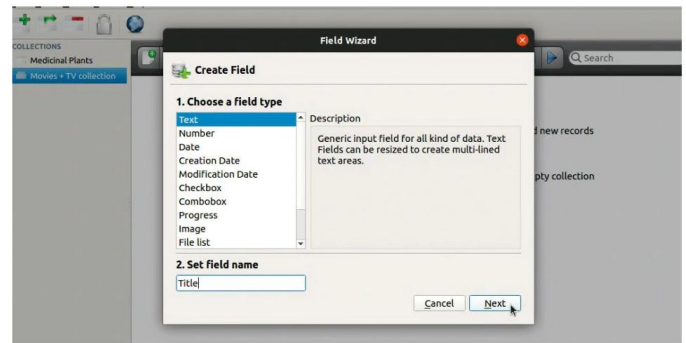
After setting it up, switch to the External Data tab to import data from a file (the CSV you exported from *Symphytum*). The default import settings should be fine, but note that some field types (such as image and file) aren't directly supported. All fields are imported as plain text unless you specify differently during the import.

*Kexi* has several advantages over *Symphytum* in addition to its support for relational databases. There's a more powerful search tool that can be restricted to a specific field and it also supports partial as well as exact matches. There's also a freeform Form designer tool and support for connecting to database servers as well as working with simple file-based databases.

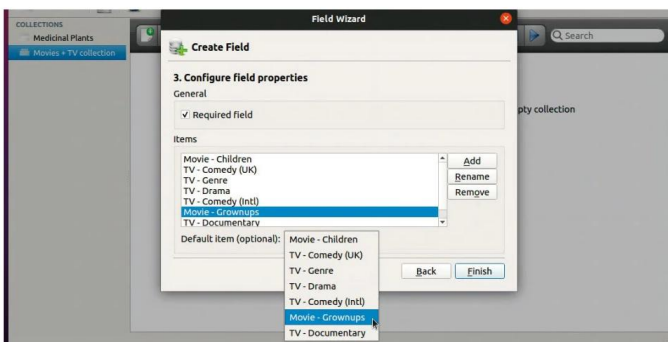


**SET UP YOUR FIRST DATABASE****1 Create a new database**

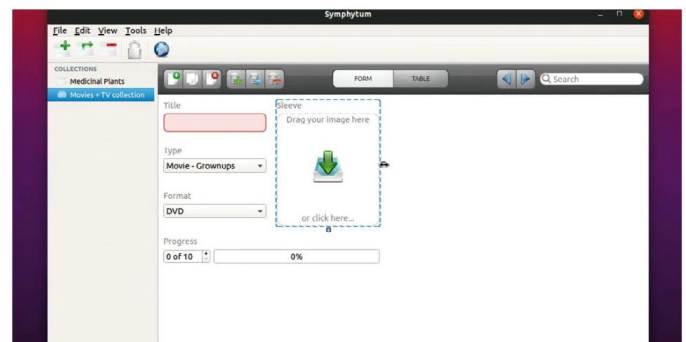
To create a new database from scratch, simply click the Create a new collection button beneath the File menu. The new database appears in the left-hand sidebar, so start by giving it a suitable name. You'll start in Form view, with a couple of handy shortcuts highlighted. Start by clicking the New Field button to define the first field of your database.

**2 Set up your first field**

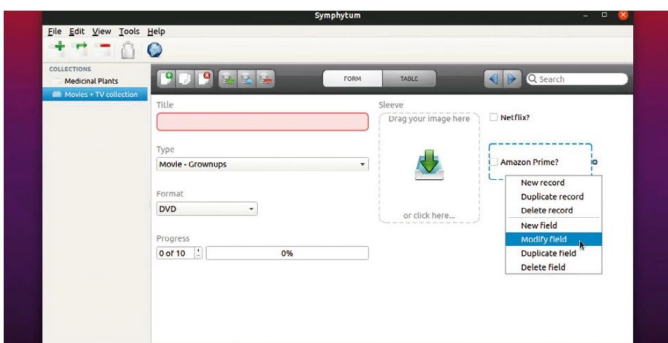
When you press the New Field button, the Field Wizard pops up. First, define what type of field you wish to create from a choice of 12 (see the main text) – select one for a description of what it does. Once you've set your field type, give it a name such as Title and click Next. If the field must always be filled in, tick Required field before clicking Finish.

**3 Create subsequent fields**

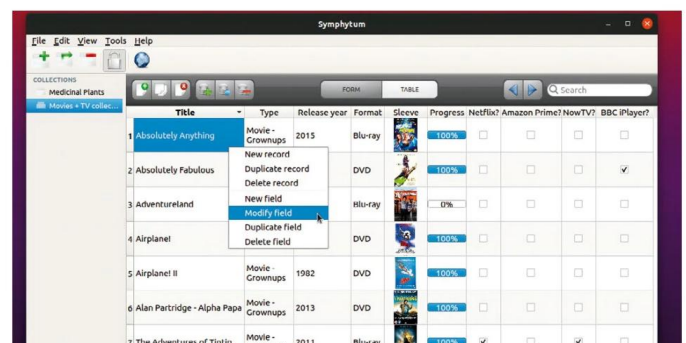
Repeat the process to create all the fields you think your database needs (it's worth sketching these on paper before you begin). Certain field types will trigger more context-sensitive options within the wizard – for example, choose Combobox and you'll be prompted to enter each choice you wish to appear in the drop-down menu.

**4 Redesign your form**

Once your fields are in place, click the New Record button or press Ctrl+N. The Form will switch to show all the fields in your database, dynamically positioned on-screen. To rearrange their order, just click a field's title to reveal its borders. You can click and drag to move it, or use the drag handles on the right and bottom to resize the field as required.

**5 Modify or remove fields**

You can also modify and remove fields when in Form view proper. Right-click the field to modify it by changing its name and associated options with the selected field type (but note that you can't change the field type). Other options include deleting the field or creating a duplicate based on it (you'll be prompted to change the field name).

**6 Edit in table view**

You can also add, modify, and remove fields in Table View. Just use the New Field button to create a new field from scratch in the usual way, or right-click an existing field column to access options for modifying, duplicating or deleting it. Once created, switch back to Form view to arrange the new field's position as required on-screen.



Credit: www.mediagoblin.org

## MEDIAGOBLIN

# Build a streaming video web service

**Mike McCallister** helps you collect and share your children's sporting exploits with their teammates with the free video service MediaGoblin.



**OUR  
EXPERT**

**Mike McCallister** is more interested in watching sports than playing them, but loves the open web.

## QUICK TIP

When building MediaGoblin on Debian, first add export `PATH="$PATH:/usr/sbin"` to your `.bashrc` file to enable using `sudo`.

Here's a section of the mediagoblin.ini file.

```
[mediagoblin]
direct_remote_path = /mgoblin_static/
email_sender_address = "admin@brixtonysa.org"
```

**T**he (fictional) Brixton Youth Sports Association (BYSA) needs a place to store instructional videos, game footage, schedules and all the other things to celebrate and educate players, coaches and parents in the sports they organise. They could use YouTube or any other commercial video site, but it's better to have something a bit more private, no?

Enter *MediaGoblin*, a Python-based tool enabling folks to share all kinds of media with just the people they want to share with, no corporate servers required. *MediaGoblin* uses the ActivityPub web standard (see box, bottom right) to permit sharing.

## Enjoying MediaGoblin

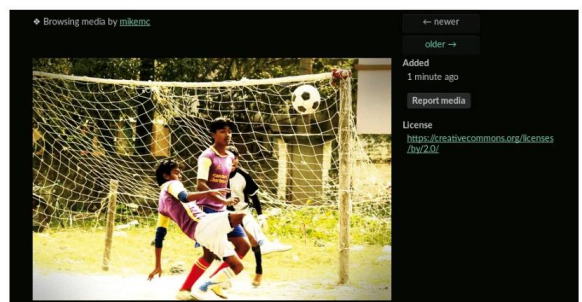
*MediaGoblin* doesn't yet offer the prettiest site on the internet (if you're so inclined, you can create your own theme), but it works pretty well.

Users start by setting up a simple profile with a username, password and email address. After confirming the email address, they can add a bio if they like. Viewing media is just a click away.

Click Add Media to share the latest content. Browse to the file and add a Title and Description. You can add tags to your upload to make it easier to find. The license menu offers a variety of Creative Commons licenses and a Public Domain option as well as All Rights Reserved. Click Add to finish. Depending on how the site is set up, *MediaGoblin* supports uploading and viewing still images, audio, video, PDF documents, presentations and 3D models. You can also add comments to each video or other posted items. Before you can do all of this, someone has to set up the server. Here's how...

## Setup prerequisites

You can set up *MediaGoblin* on any machine connected to the web. You'll need enough storage capacity to hold a lot of content. You can use Apache or any other web server. The developers make clear that they prefer the nginx web server, especially for production sites. The developers provide complete and tested deployment



*MediaGoblin* gives your youth sports group a place to store and share videos, photos, documents and other media. You control it, too.

instructions for Fedora and Debian at <https://bit.ly/LXF267-MGdocs>.

To make things happen, make sure all these packages are installed on your system:

- > Install git to work with the latest *MediaGoblin* source
- > For building *MediaGoblin* from source: *automake*, *gcc*, *make*
- > Python 3.4: *python3-dev*, *python3-gi*, *python3-gst-1.0*, *python3-lxml*, *python3-pil*, *python3-pillow*
- > Javascript: *nodejs*, *npm*
- > Virtualenv

For more industrial-sized use, you'll want to use nginx, RabbitMQ and PostgreSQL, but for our sports league, SQLite will work just fine for a database.

## Make an unprivileged user

What makes sharing easy but secure in *MediaGoblin* is that it's owned by an unprivileged system user. All we have to do is create a system account for *MediaGoblin*. Open a user manager, or use this shell command:

```
sudo useradd --system --create-home --home-dir var/lib/qmediagoblin --group <groupname> --comment 'GNU MediaGoblin system comment' mediagoblin
```

Add this user to an existing group associated with the web server, so the server can read uploaded media files. Depending on your distribution, this could be `www-data` or something similar. Other distributions may not allow a comment.

## Where do the files go?

You don't want the BYSA files mixed up with the rest of the files on your system, so let's make a directory. This

Credit: Pabak Sarkar, CC BY 2.0.  
www.flickr.com/photos/pabak/14181670605/



directory is for both the MediaGoblin program files that come from Git, and the shared media files. So if the BYSA website is [brixtonysa.org](https://brixtonysa.org), add this directory:

```
sudo mkdir -parents /srv/mediagoblin.brixtonysa.org
sudo chown -nodereference -recursive
mediagoblin:www-data /srv/mediagoblin.brixtonysa.org
```

## Let's install!

You have to change identities now to install and work with the new unprivileged *MediaGoblin* user. Type

```
sudo su mediagoblin -shell=/bin/bash
```

Change to the new directory you created, **/srv/mediagoblin.brixtonysa.org** and pull in the *MediaGoblin* files from git:

```
git clone --depth=1 https://git.savannah.gnu.org/git/mediagoblin.git --branch stable > --recursive
```

The current stable *MediaGoblin* files populate the directory, and (among other things) creates another *MediaGoblin* subdirectory. Change to that directory and run **.bootstrap.sh**. Next, set up virtualenv:

```
VIRTUALENV_FLAGS='- systemsite-packages' ./configure.
```

Then run **make** to build *MediaGoblin*. The shared media files will be stored in the **user\_dev** directory. Set the proper permissions:

```
mkdir -mode=2750 user_dev
```

## Tweaking the Database

Before you run *MediaGoblin* for the first time, change the default configuration file to update **email\_sender\_address** to direct *MediaGoblin*'s system emails to the right place. To ensure that new account confirmation emails go out, also set **email\_debug\_mode = false**.

Populate the SQLite database. Back to the shell:

```
bin/gmg dbupdate
```

Make a database admin user. Where it says "you," enter a username and real email address. Enter a secure password when asked:

```
./bin/gmg adduser --username <you> --email <you@example.com>
```

```
./bin/gmg makeadmin <you>
```

Create a space for your log file in **/var/log/mediagoblin**. You should now be ready to test your setup. Run this command, and open your browser to **localhost:6543**:

```
./lazyserver.sh --server-name=broadcast
```

When you've confirmed that *MediaGoblin* is running on your server, you can then set up as a Systemd service to put the site on 24/7.

Go to <https://mediagoblin.readthedocs.io> (<https://bit.ly/LXF267-MGdocs>) and click the Deploying MediaGoblin page. Towards the bottom, you'll see "Run MediaGoblin as a system service". You'll see a copy of the Celery and Paste service configuration files. Edit the directory paths to match your site and save each of them into **/etc/systemd/system**.

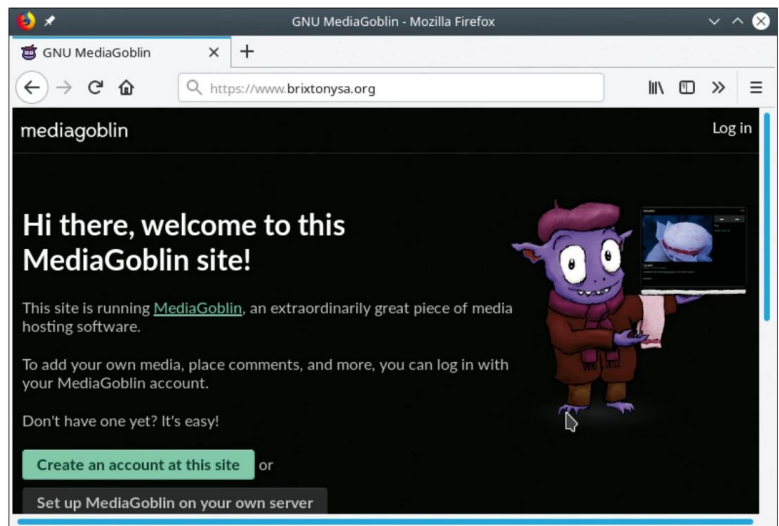
Back in the shell, start the MediaGoblin service:

```
sudo systemctl enable mediagoblin-celeryd.service &&
sudo systemctl enable mediagoblin-paster.service
```

## Making things attractive

As the admin of [brixtonysa.org](https://brixtonysa.org), you'll want to help your users get started sharing files on the site.

Since the Brixton Youth Sports Association manages activities in several sports, you may want to organise



the media in multiple Collections, one for each sport. Click Create New Collection and add a title and description (Football, Basketball, Swimming and so on.).

If you've got some content ready for people to look at before the launch, you can use the site's upload tools to add, but as an admin, you can also use the command line to upload one or 1,000 files to the database.

For a single file, type **./bin/gmg addmedia <your username> <filename>**. You can add metadata by adding **--title**, **--description**, **--collection-slug**, **--license** and the like. You can do more by creating a metadata file with this syntax: **Location , dc:title , dc:creator , dc:description , dc:collection-slug , dc:type**.

Here, the location can be on the local machine or has a URL, and creator is your username. Each item will also be enclosed with quotes: **"Frankie makes a goal"**. Learn more on the Command-line uploading page in the Admin Guide.

## Hoarding your media

*MediaGoblin* makes it easy for anyone with a need to share media with a group of folks while protecting privacy from the rest of the world. Perfect for coaching young people to get active and get better at their chosen sport.

Now build that system with a few terabytes of storage space, and get going on filling them! **LXF**

Click Recent Media from the home page to see the new stuff. The page includes more information.

## » THE ACTIVITYPUB PROJECT

The ActivityPub web standard came about because some developers got sick of social activity moving to walled gardens like Facebook, YouTube and Twitter. Communication and collaboration was built into the World Wide Web at its birth! Supporters of ActivityPub describe the standard as "provid(ing) a client to server API for creating, updating and deleting content, as well as a federated server to server API for delivering notifications and subscribing to content."

Today ActivityPub powers the Social Web, sometimes called the Fediverse for the federated connections between independent websites. Developers are using the standard to power a variety of alternatives for organizing events, sharing music, and just hanging out. These include Mastodon (microblogging), PeerTube (video sharing), and Discourse (discussion boards).

Learn more about ActivityPub at [www.activitypub.rocks](https://www.activitypub.rocks).



# GET LINUX FORMAT FROM **£4.17** A MONTH



This offer is **ONLY** available via our online shop  
[www.myfavouritemagazines.co.uk/lind/DIGCSP20](http://www.myfavouritemagazines.co.uk/lind/DIGCSP20)

Available on iOS and Android



# HotPicks

Symphonium » Cadmus » E4head » Gromit-MPX  
 » Fsmon » Qt-SESAM » Video-trimmer » Pillar Valley  
 » Caveexpress » Flatpak » Viper Browser



**Alexander Tolstoy**

has an open source app for everything, not just for playing arcade games from yesteryear.

## PIANO TRAINER

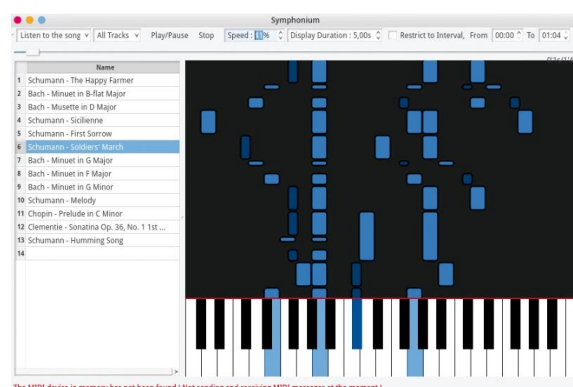
# Symphonium

**Version:** 0.1 **Web:** <https://github.com/ttdm/symphonium>

**M**ost sound recording software include a digital piano keyboard, designed to work as a virtual MIDI device. If you haven't tried one, there are numerous YouTube lessons available that will show you how to play popular tunes, by pressing the right keys in the right order, for the correct amount of time. These usually take the form of a standard piano keyboard located along the bottom of the screen, with the area above it featuring falling elements that represent the keys a user needs to press. This beginner-friendly way of learning is available as an open source application.

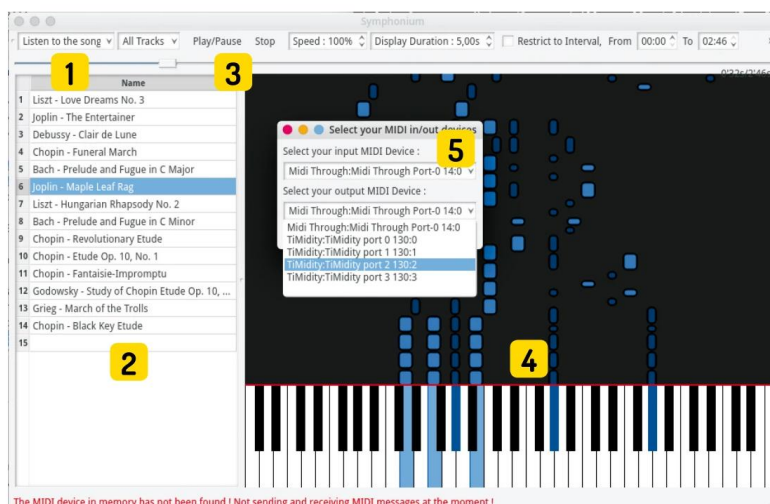
*Symphonium* is a piano trainer that plays MIDI files. It has a clear graphical interface and is ideal for learning. The program is in an early state of development, but it's already very usable. Load a file or a directory with MIDI files, hit Play and see how it goes. You can change the tempo, display duration for the graph, and restrict the file to a given interval. *Symphonium* works in three modes – listen, practice and play – which means that for anything other than listening a user needs an input MIDI device. However, if one isn't available the application still proves useful. The only obstacle is that many Linux distros don't come correctly configured to play MIDI files out of the box, especially if there's no MIDI hardware. For that reason *Symphonium* will load up and play files with no issues, but without sound. So make sure that you have the Timidity server up and running before launching the application.

Timidity adds the required virtual MIDI devices that can be used via the *Symphonium* settings (Options>Select MIDI device). Once the sound is working, supply the application with a MIDI file. *Symphonium* already comes bundled with a decent collection of classics for intermediate and advanced levels, made up of 43 samples, and it accepts virtually any correct MIDI file – even if it were created for a different instrument. That said, with *Symphonium* you can find out how any instrument part can be converted to become a piano part!



Sometimes it makes sense to slow down the tempo – it gives you a chance to follow the tune more closely.

## Getting to know Symphonium...



**1 Choose the right app mode**  
 By default *Symphonium* is in the Listening mode, but if you have an input MIDI device, choose either Learn or Play mode.

**2 Manage the playlist**  
*Symphonium* comes with a great collection of classic tunes, but you can always import custom MIDI files.

**3 Finegrain controls**  
 Use the main toolbar to control the

playback, change speed, display duration, and set the interval.

**4 Main working area**  
 This area above the keyboard helps you to understand what piano keys you need to press, and when.

**5 Set up MIDI devices**  
 If you hear nothing or are unable to change the app mode, use these settings to manually select the right MIDI devices.



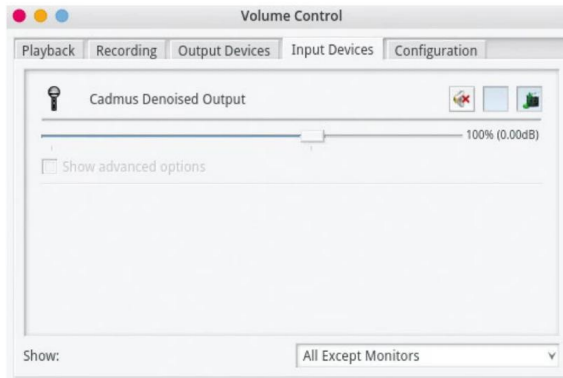
## NOISE REMOVER

# Cadmus

Version: 0.0.1 Web: <https://github.com/josh-richardson/cadmus>

**F**ew people want background noise in their audio recordings, and there are several open source tools to eliminate it. If it involves post-processing an audio file that's already been recorded, then we'll probably plump for *Audacity*, but what if we need to suppress noise in real time? Gamers who maintain voice chats or client desk workers who make announcements in public places are just a few of the possible use cases. Some proprietary chatting clients such as *Teamspeak* have limited tools for muting undesired sounds, but an easy-to-use solution was yet to become available... until now.

*Cadmus* provides a front-end to the LADSPA Real-time Noise Suppression plug-in (<https://github.com/werman/noise-suppression-for-voice>). The latter is a wrapper for Xiph's RNNoise library, which applies a recurrent neural net to detect and mute background sounds. So, in this chain of three elements, *Cadmus* is the most newbie-friendly one. It's a PyQt5-based application that comes bundled with all the required internals (including `librnnoise_ladspa.so`) and provides



Cadmus creates a denoised output device that should sound better than the standard one.

a system tray icon with a simple context menu that supports *Pulseaudio*. First, select an input device (a microphone) and then find two new *Pulseaudio* devices in your mixer: one for output (monitor), and another for the denoised microphone input device. Once you've set up the devices correctly, the microphone should perform a lot better, without background or ambient sounds, hiss, clicks and all other sorts of undesired noise. The RNNoise back-end that *Cadmus* relies on does a good job in crowded places, and it shines at reducing mechanical keyboard clicks, noisy fans and so on. No setup is required, and changes are applied once *Cadmus* is enabled.

The project download pages contains the AppImage, the static build in the ZIP archive, and of course the source tree release, for those who prefer to build and install Python packages via *Pip*.

## PERFORMANCE OPTIMISATION TOOL

# E4head

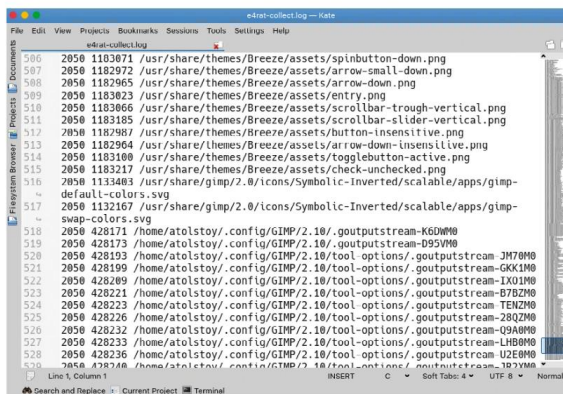
Version: GIT Web: <https://github.com/leandromqrs/e4head>

**T**his article was written when Fedora developers were discussing the proposal to switch from LVM+Ext4 to Btrfs as the default choice of filesystem during installation. Without going too deep into comparing the pros and cons of both filesystems, let's point out a promising and attractive Ext4-specific tool. *E4head* is the reincarnation of the older *E4rat* application, or *Ext4 Reduced Access Time*. The idea is to make use of the online defragmentation `EXT4_IOC_MOVE_EXT ioctl` available exclusively in that filesystem to reallocate frequently used data. With Ext4 being already super-fast, *E4head* allows extra optimisation by a noticeable margin, which is a great tweak for I/O intensive systems and low-end configurations.

Normally, *E4head* works the same as the original *E4rat*. It implies three phases: first for 'learning' (collecting files); second for reallocating what's been collected; and third for preloading reallocated data to page cache. Without further ado, let's do all three steps:

```
$ sudo systemctl stop auditd
```

We do this as it conflicts with *E4rat*.



Before reallocating, make sure that there's enough data collected by `e4rat-collect`.

```
$ sudo e4rat-collect // (start opening apps you want to optimize, hit Ctrl+C when done)
```

```
$ sudo e4rat-realloc e4rat-collect.log && sudo e4rat-realloc e4rat-collect.log
```

You can examine `e4rat-collect.log` to find out how many files were involved and how many fragmented chunks they have. You can also run *E4rat* by adding `init=sbin/e4rat-collect` to your Grub boot options and then examining the collected list in `/var/lib/e4rat`.

Reallocating reduces boot time and makes applications load faster. Actual results vary depending on many factors, including filesystem fragmentation. Use the following utility (from the *E2fsprogs* package) to find out the level of fragmentation:

```
$ sudo e2fsck /dev/sdX
```

If you see many small-sized extents, then *E4head* will be very effective (but don't forget `e4defrag`, too!)



# Gromit-MPX

[illegible]

lot on-screen. The software is easy to use, and it'll take seconds to get used to on-screen drawing. However, *Gromit-MPX* can do more than just paint with a thin red brush. Edit `~/config/gromit-mpx.ini` to set up extra tools (marker or eraser, say), custom colours, brush sizes and opacity. The *Gromit-MPX* source archive provides a sample configuration file as a template for such modifications. Finally, the tray icon is also helpful because it reveals the context menu with helpful tools. Use it to change brush size, opacity and toggle modes without touching any configuration files.

*Gromit-MPX* proved itself to be a handy tool with a strong focus on performance and ease of use. The built-in toggling feature enables you to draw on-screen, then disable the drawing visibility and regain control over the desktop, and after that at any moment bring the drawing back.

# Fsmon

```
File Edit View Bookmarks Settings Help
- Studio fishman - Konsole

PSE_STAT_CHANGED 13610 /home/atolstoy/Source
PSE_OPEN 13610 /home/atolstoy/Source
PSE_STAT_CHANGED 13610 /home/atolstoy/Source
PSE_STAT_CHANGED 13610 /home/atolstoy/Source
PSE_OPEN 2209 /home/atolstoy/.cache/thumbnailsnormal/4Adf56509c4b40
PSE_STAT_CHANGED 2209 /home/atolstoy/.cache/thumbnailsnormal/4Adf5650
PSE_STAT_CHANGED 2209 /home/atolstoy/.cache/thumbnailsnormal/4Adf5650
PSE_OPEN 2209 /home/atolstoy/.cache/thumbnailsnormal/aes179cGf4e2938
PSE_STAT_CHANGED 2209 /home/atolstoy/.cache/thumbnailsnormal/aes179cGf4e2938
PSE_STAT_CHANGED 2209 /home/atolstoy/.cache/thumbnailsnormal/aes179cGf4e2938
PSE_OPEN 2209 /home/atolstoy/.cache/thumbnailsnormal/aes179cGf4e2938
PSE_STAT_CHANGED 2209 /home/atolstoy/.cache/thumbnailsnormal/aes179cGf4e2938
PSE_STAT_CHANGED 13610 /home/atolstoy/1
PSE_STAT_CHANGED 13610 /home/atolstoy/1
PSE_STAT_CHANGED 13610 /home/atolstoy/1
PSE_CONTENT_MODIFIED 13571 /home/atolstoy/.config/konsolerc.Lock
PSE_OPEN 13571 /home/atolstoy/.config/konsolerc
PSE_STAT_CHANGED 13571 /home/atolstoy/.config/konsolerc
PSE_OPEN 13571 /home/atolstoy/.config/konsolerc
PSE_CONTENT_MODIFIED 13571 /home/atolstoy/.config/konsolerc (deleted)
PSE_CREATE_FILE 13571 /home/atolstoy/.config/konsolerc (deleted)
PSE_OPEN 13571 /home/atolstoy/.config/konsolerc (deleted)
PSE_CONTENT_MODIFIED 13571 /home/atolstoy/.config/konsolerc.Lock
PSE_OPEN 13571 /home/atolstoy/.config/konsolerc
PSE_STAT_CHANGED 13571 /home/atolstoy/.config/konsolerc
PSE_CONTENT_MODIFIED 13571 /home/atolstoy/.config/konsolerc (deleted)
PSE_CREATE_FILE 13571 /home/atolstoy/.config/konsolerc (deleted)
PSE_DELETE_FILE 13571 /home/atolstoy/.config/konsolerc.Lock
```

Even if your Linux OS looks idle, there may be a lot of hidden activity on the disk.

several monitoring back-ends, including *kdebug* (for Apple's XNU kernel and BSD kernel debug), *fanotify*, *inotify* and few more (check the list with `$ fsmon -L`).

The use cases that benefit from *Fsmon* can differ widely. What comes to mind first is application development debug of any kind, including Android and iOS projects. Next, you may want to monitor certain directories and quickly detect any penetration or abnormal activity there, like, for example, on a remote host. *Fsmon* is ideal for finding out what generates so much I/O in case your system processes monitor fails to provide answers. If you start a program and want to know which configuration files it tries to source, then *Fsmon* is also the right thing to use. Finally, you can limit the flow of the events by instructing *Fsmon* to monitor just the provided PID (and optionally its children too). See `$ fsmon -h` for help and examples.

Now run the tool for any target, for example the home directory (**\$ fsmon ~**) and then do something trivial, such as open a file manager or refresh a web page. Notice how much lower-level activity takes place! *Fsmon* outputs everything it detects with PIDs, process names and paths for each entry. Note that the tool comes with



## PASSWORD GENERATOR

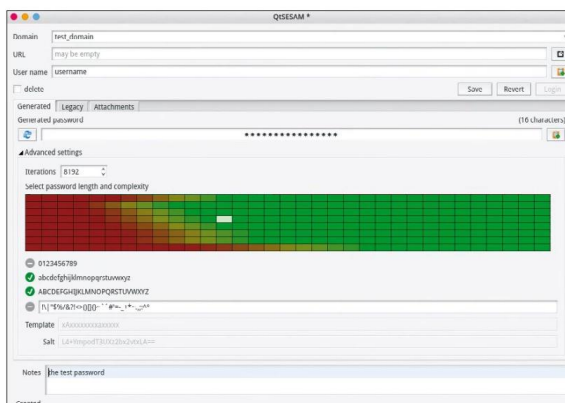
## Qt-SESAM

Version: 2.1-beta Web: <https://github.com/ola-ct/Qt-SESAM>

Every once in a while there's a study on network security breaches, of which a significant part is usually made up of trivial things, such as admin:password credentials for home Wi-Fi routers. One way to avoid such vulnerabilities is to use strong passwords whenever possible. The Linux ecosystem already has a few decent tools to generate and store good passwords, such as *Keepassxc* (see **LXF243**). However, this time we've got another privacy aware tool that boasts even stronger focus on password quality.

Qt-SESAM is a front-end to *Super-Easy & Secure Authentication Manager* (SESAM) and it enables users to generate and manage passwords for any service, such as websites, accounts or applications, without having a master degree in cryptography. From a typical Linux user who has some experience in compiling open source tools from source, Qt-SESAM is a utility that's easier to install than to use. Therefore, a little extra knowledge is required to get started.

The UI assumes that you can supply a URL, a user name and log-in using the generated password.



Picking a strong password assumes that green is good and red... isn't.

Furthermore, you can share your tool's settings with a remote sync server, or a file that, again, can either be local or available from a cloud service. All those complexities are completely arbitrary and don't prevent anyone from using Qt-SESAM solely to produce strong passwords. The tool makes use of 256-bit AES keys, spiced with randomly generated salt data. Although it's already secure enough in our view, Qt-SESAM locks everything with a master password that you need to provide upon the first launch.

When Qt-SESAM is running, it asks you to provide that password after some time of inactivity. And, of course, the most advertised feature is a graphical password strength chooser. Reveal the Advanced Settings panel and identify a coloured widget with red, yellow and green squares. Use it to select a high-quality password (green) and avoid short and weak passwords (red).

## VIDEO CUTTER

## Video-trimmer

Version: 0.2.0 Web: <https://gitlab.gnome.org/YaLTeR/video-trimmer>

Normally, few people would reach for a heavyweight video-editing application such as *Kdenlive* only to trim a piece of footage. There are plethora of smaller, faster, open source programs that can do this better. The one that caught our attention this month is *Video-trimmer*. This is a GTK3-based application that perfectly complements the Gnome3 desktop design approach, although you're free to use the program elsewhere.

*Video-trimmer* is simple to use. Load the video file that you need to trim, cut off something from the beginning or near the end, and use the selection bar below the picture to set the range. There are also two text input fields for more precise control below the bar. Drag the range boundaries, or provide text time stamps and once done, hit the Trim button to apply changes and then save the result.

*Video-trimmer* is written in Rust and powered by *Gstreamer* and *FFmpeg*. It supports any video that will play in Gnome's stock video player (aka Totem). As for resource usage, we should point out that trimming does

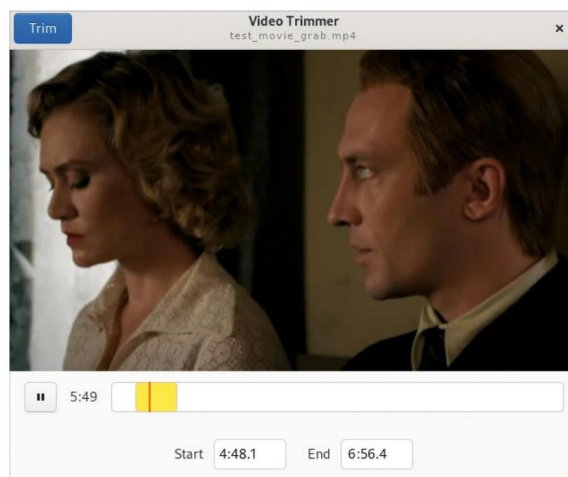
not involve re-encoding – it's performed in-place. That also means that even if the input video file is huge, trimming

will still be a quick job. Of course, the same feature is accessible in the command-line tool *ffmpeg*, but a clear GUI with a strong focus on some designated feature is often preferable. *Video-trimmer* is ideal for truncating videos stored in shared network folder, provided that you use an appropriate file manager, such as *Nautilus*.

Getting *Video-trimmer* to run is simple as well. For those missing the classic build-and-install way, here's the command sequence:

```
$ meson -c build && ninja -c build
$ sudo ninja -c build install
```

If you record your own videos or edit cuts using a GTK-based Linux desktop, consider adding *Video-trimmer* to your list of handy utilities.



Cutting a small sequence from a lengthy film takes only few seconds with this tool.



**SKILLS-BASED GAME**

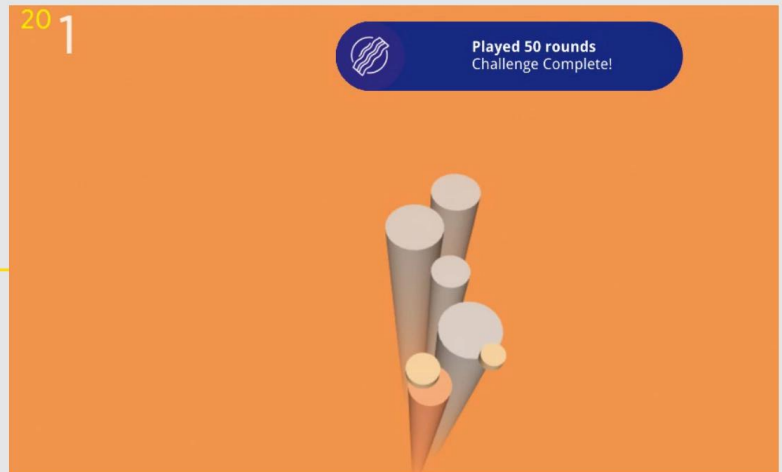
# Pillar Valley

**Version:** GIT **Web:** <https://github.com/EvanBacon/pillar-valley>

**T**he *Pillar Valley* author declares on its Github front page that, "It's time to immerse ourselves in a suave world of zen!" So, we took the trendy mobile game for a test ride in a desktop web browser and found it utterly fun, captivating and worth spending more than a few minutes of your time playing.

*Pillar Valley* is a minimalist game that features abstract vector objects – there's no bitmap artwork on show. The game features tall pillars that you need to land a revolving tablet on. Essentially, it's a challenge that involves good reaction times. You're given three attempts for each pillar because the tablet is constantly shrinking and it can manage only three rounds before it disappears. The tablet size, the revolving radius and the rolling direction randomly change, and the more pillars you complete, the harder it gets.

The game has a range of prizes, which pop up once you complete a challenge (play some rounds, set a record and so on). Pillars are infinite, and the game is virtually endless. If you fail, it takes you to the beginning and displays your highscore in the top-left corner.



Pillar Valley was originally intended to run on Android and iOS, and this can be easily seen in the source tree. However, because the game was made using the Three.js and Expo frameworks, it's also possible to run a local server. Make sure you have *Yarn* installed (look it up in your package manager), grab the code and run `$ yarn` in the 'client' subdirectory. That will install all required dependencies, including Expo. After that, run `$ expo start` and give it some time to compile the code and fire up the local server. Once that's done, the game should be available at <http://localhost:19006> or similar port. Play it with no limitations, ads or other annoyances right on your desktop Linux machine!

Try to jump from one stone to another – and there's zero margin for error!

**PLATFORM GAME**

# Caveexpress

**Version:** 2.5.1 **Web:** <https://github.com/mgerhardy/caveexpress>

**M**ost platform-style arcade games have so much in common that it's a real challenge for developers to make their games original, addictive and fun to play. So far we've discovered quite a few high-quality platformers that you can build and run in Linux with just a little effort. Perhaps the most colourful and well-made was *Frogatto* (**LXF227**), and therefore we were happy to find out that *Caveexpress* is a decent rival to it in terms of graphics, game design and mechanics.

The game takes us back to pseudo-prehistoric times when people lived in caves and had to avoid dinosaurs, mammoths and giant fish! However, *Caveexpress* is more about delivering goods and passengers than survival. Our hero is a cavemen who controls a pedal-driven flying machine made of sticks, leaves and ropes. He has to pick up crates and transfer them to a shredding station (the game has some odd terminology).

Flying around a level requires a good amount of control over momentum – indeed, the game's physics are similar to those found in *Flappy Bird*.



The caveman needs to avoid enemies and also take care of falling from a high altitude. When the health meter is running low, the cavemen can pick up a stone and throw it at a tree. Doing this dislodges fruit from the tree that you can pick up and eat, which restores health.

*Caveexpress* is very playable in a way that it showcases different arcade styles in one game. Levels may have various objectives: collect crates, deliver something in time, transfer passengers, finding a way through rocks, water and other elements, and so on. The artwork is particularly eye-catching, and the music generates a perfect stone-age feeling. The game has a useful tutorial that helps you complete basic tasks before letting you go out into the wild world. The angry mastodons, terrifying pterodactyls and others would rather see you become extinct, but you can always hit them from above with a stone!

If the pterodactyl looks happy then you'll probably need to start the level again!



## SOFTWARE DEPLOYMENT UTILITY

# Flatpak

Version: 1.8.1 Web: <https://flatpak.org>

**T**he two main rivals in application sandboxing are *Snap* and *Flatpak* (although *Applmage* is still around). One is made by Canonical and another by Red Hat, but these days it doesn't lead to many limitations because you can use both technologies in almost any Linux distro. For instance, we picked Bauh in **LXF266** to manage all Snaps, Flatpaks and Applimages on a Linux-powered machine.

Although a back-end tool, *Flatpak* boasts a fast rate of development. Together with the fact that containers are still perceived as a next-generation computing development, this makes users feel like they're using cutting-edge technology. Tastes may differ, and some people prefer to stick to old-school package managers, but *Flatpak* definitely brings more choice and freedom to how we get hold of applications. For instance, *Flatpak* is ideal for trying out the latest versions of such popular software like *LibreOffice*, *Gimp* and *Inkscape*, all available from [www.flathub.org](http://www.flathub.org).

We tested *Flatpak* 1.8 and discovered some great additions and fixes. Filesystem permissions setup now enables you to access a host's `/usr`, `/etc` and `/lib`

```

atolstoy@opensuse-Main:~$ LANG=C flatpak --help
Usage:
  flatpak [OPTION...] COMMAND

Built-in Commands:
  Manage installed applications and runtimes
  install          Install an application or runtime
  update          Update an installed application or runtime
  uninstall       Uninstall an installed application or runtime
  mask            Mask out updates and automatic installation
  list            List installed apps and/or runtimes
  info            Show info for installed app or runtime
  history         Show history
  config          Configure flatpak
  repair          Repair flatpak installation
  create-usb      Put applications or runtimes onto removable media

Finding applications and runtimes
  search          Search for remote apps/runtimes
  
```

Many software stores in Linux work with Flatpaks just fine, without any glitches.

directories from within a sandboxed application. Time syncing now takes the host's `/etc/localtime` file into account, programs can now directly access ALSA sound systems if *Pulseaudio* was allowed, and so on.

It's good to hear that Flatpak developers conduct tests and don't just assume that if a program successfully starts, then it works. Such use cases as recording audio data from a sandboxed *Audacity* or attaching files to a forum post in a sandboxed browser show that *Flatpak* is becoming more user-friendly. Perhaps in the future many Linux users may not even notice that they're using *Flatpak* versions of their applications. Much as it already works in GNOME Software in Fedora, where *Flatpak* is suggested as a default option once a choice becomes available.

## WEB BROWSER

# Viper Browser

Version: GIT Web: <https://github.com/LeFroid/Viper-Browser>

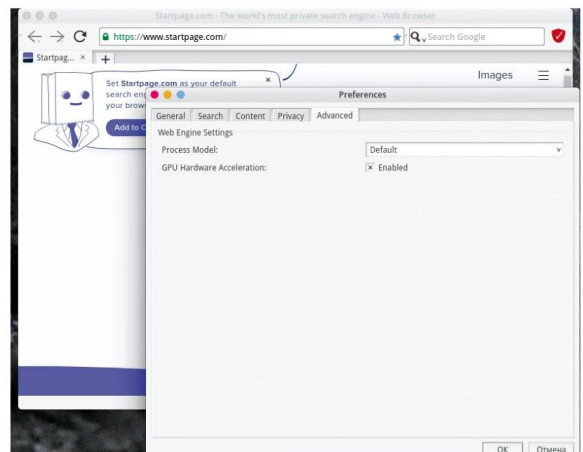
**B**rowsing the web is one computing task that's well served by open source software. This time we're bringing even more choice to Linux users who value speed, privacy and the minimalist look of such browsers as *Falkon* and *Otter Browser*. This month's pick is called *Viper Browser*, and it showcases an alternative vision of what a perfect WebEngine-based browser should be like.

*Viper Browser* has a range of distinct features that make it stand out. The first-run wizard asks users some questions and enables them to choose a search provider, customise ad blocking filters (for example, add uBlock lists), optionally enable the built-in *PDF.js* viewer and Flash, send "Do not track" to all visited pages and even suppresses those cookie accept warnings. The program's advanced features and settings are just a couple of mouse clicks away, whereas other similar browsers tend to hide these items away from users.

*Viper Browser's* settings also stand out, giving users greater control over WebEngine settings (that's the modified *Chromium* code wrapped as a Qt component).

In Preferences>Advanced you can change the process model from Process per website to Single process, ensuring that *Viper Browser* behaves just the way you want it to.

Surfing the web with *Viper Browser* was generally a pleasant experience. Given the fact that this is a typical underdog browser – for instance, there's no extensions store – *Viper* still looks fresh, original and promising. It's also great to have advanced rendering settings close at hand, and other common features implemented in a more convenient way. For example, *Viper's* UI for changing the default search provider is far more logical than that found in *Otter Browser*. As you'd expect, *Viper Browser* also has tabs, history, a full-screen switcher and a download manager with pause button. Recommended for those looking for a decent alternative web browser. **LXF**



Toggle GPU acceleration and engine process model with just a few mouse clicks!



# BACK ISSUES >> MISSED ONE?

## ISSUE 266 Summer 2020

Product code:  
**LXFDB0266**



### In the magazine

We see what's new in the latest version of Mint, narrow down the best IDE for your programming adventures, show you how to code the arcade classic *Pong*, manage tasks using *Zenkit To Do*, and speak about the Open Mainframe Project with John Mertic.

### Digital download highlights

OpenSUSE 15.2 (64-bit) and Grml 2020.06 (32- and 64-bit versions).

## ISSUE 265 August 2020

Product code:  
**LXFDB0265**



### In the magazine

We check out the latest show-stopping distros that are packed with advanced features and tools, show how to code *Pac-Man* games in Python, design and publish eBooks, create stop-motion animation on a Pi, and explore modern browsers.

### Digital download highlights

Linux Lite 5.0 and Pop!\_OS 20.04 (both 64-bit versions).

## ISSUE 264 July 2020

Product code:  
**LXFDB0264**



### In the magazine

We reveal why the latest version of Ubuntu is better than ever before, show you how to code a top-down arcade shooter, improve your audio recordings with *Audacity*, simulate your own circuits and give you an insight into open source processors.

### Digital download highlights

Fedora 32 Workstation and Manjaro 20 (both 64-bit versions).

## ISSUE 263 June 2020

Product code:  
**LXFDB0263**



### In the magazine

Discover how you can bring your old hardware back to life with 32-bit distros, and code an old-skool platformer. Plus, secure your cloud backups, solve mazes using graph algorithms and learn new video-editing skills with the very capable *OpenShot*.

### Digital download highlights

Mint Debian Edition 4, antiX 19 and Bodhi Linux 5.1 (all 32-bit versions).

## ISSUE 262 May 2020

Product code:  
**LXFDB0262**



### In the magazine

We show how to keep your Linux system safe, from protecting your data to rock-solid login solutions. There's advice on how to construct an FM radio receiver, make the most of system-monitoring tools, and a look back at over 20 long years of *Linux Format*!

### DVD highlights

Manjaro 19.0.1 (64-bit) and IPFire 2.25 Core 141 firewall distro (32- and 64-bit).

## ISSUE 261 April 2020

Product code:  
**LXFDB0261**



### In the magazine

We take an in-depth look at virtualisation to ease our installing woes. We also explore the world of FreeBSD. Plus, learn how to use *LibreOffice Calc* for serious mathematical work, and how *Zabbix* can solve an enterprise's monitoring needs.

### DVD highlights

Solus 4.1 (64-bit), as well as OpenMandriva 4.1 (64-bit).

# To order, visit [myfavouritemagazines.co.uk](http://myfavouritemagazines.co.uk)

Select Tech from the tabs of magazine categories, then select Linux Format.

Or call the back issues hotline on **0344 848 2852**  
or **+44 344 848 2852** for overseas orders.

Quote the Product code shown above and have your credit or debit card details ready



Credit: <https://vice-emu.sourceforge.io>

## EMULATION

# BASIC and demos on the Commodore 64

**New series**  
Don't miss  
next issue,  
subscribe on  
page 22!

**Les Pounder** reveals how to relive the 1980s with a Commodore 64 and BASIC via emulation, and without the rubbish hair and laughable fashions.



**OUR  
EXPERT**

**Les Pounder** is associate editor at Tom's Hardware and a freelance creative technologist. He blogs about his discoveries at [www.bigl.es](http://www.bigl.es).

**T**he Personal Computer with Professional Power was how the Commodore 64 was introduced back in 1982 and that power lasted for 12 years until the C64 was discontinued in 1994. But the Commodore 64 has a legacy that now spans decades.

Music, art, games and even robotics were made possible by a home computer with less power than your grandad's phone. Inside the C64 was a MOS-branded 6510 CPU running at 0.985MHz (PAL) and 1.023MHz (NTSC) along with 64KB RAM, of which 38KB was available to the user. So much was accomplished with this machine, despite now looking rather underpowered.

In the 1980s the diverse range of home computers saw groups of like-minded individuals forming clubs and user groups to support their users. Does this sound familiar to you? The storage medium of the day were cassette tapes, used to store games and programs. If you had a bit of money then a 1541 floppy drive, basically another computer that read 5¼-inch floppy disks and sent the data to the C64, could be yours for £200.

Emulating the Commodore 64 is simple, even with low-power hardware. There are a number of ways to emulate it. The first is via a commercial package from Cloanto. Called *C64 Forever* ([www.c64forever.com](http://www.c64forever.com)) this application offers easy emulation via a GUI interface. But the problem is that there's no Linux client, only Windows. To emulate the C64 on Linux we need *Vice*, which we shall install and use in this feature. There's a C64 distribution for the Raspberry Pi, called *Combian* and we looked at that in **LXF261**. The easiest way to emulate a C64 is also the most expensive. A C64 Mini console was released that provided a mini console running Linux and *Vice*. A further full-size version was also released and this had a working keyboard, but this retails for £110. That's a bit too pricey for our liking!

We used Ubuntu for our test machine, but the same instructions should also work with Raspberry Pi OS and can be adapted for other Linux distributions.

## Getting a Vice

Installing the *Vice* emulator is a little more involved than just using a package manager, but that's where we should start. Install the *Vice* emulator via the Terminal using this command

```
$ sudo apt update && sudo apt install vice
```



The Commodore 64 is almost 40 years old, and it wears those years with pride. It's responsible for many careers in the tech industry.

To use *Vice* we need a kernal, and no we've not spelt kernel wrong. In the original Commodore 64 User Guide they used "kernal". To download the kernel use the following command:

```
$ wget http://www.zimmers.net/anonftp/pub/cbm/crossplatform/emulators/VICE/vice-3.3.tar.gz
```

Extract the downloaded file and then change directory:

```
$ tar -xzf vice-3.3.tar.gz
$ cd vice-3.3/data/C64
```

Copy the kernel from the downloaded file into the *Vice* installation:

```
$ sudo cp chargen kernal basic /usr/lib/vice/C64
```

Change the directory to DRIVES and then copy files from there to the *Vice* installation directory.

```
$ cd ..
$ cd DRIVES
$ sudo cp d1541II d1571cr dos* /usr/lib/vice/DRIVES/
```

Now we can run the emulator by typing **x64** in a terminal and right now is when the nostalgia will hit you

## The basics

Home computers in the 1980s were the engines that fuelled a coding craze. Bedroom coders would spend hours writing code to create games and applications. They then shared their work with others via BBS (Bulletin Board Service) that offered places to swap games, chat and read the news. The first language that

### QUICK TIP

All of the code in this tutorial can be downloaded from our GitHub repository <https://github.com/lesp/LXF267-C64-BASIC/archive/master.zip>.



many learnt was BASIC (Beginners' All-purpose Symbolic Instruction Code), mainly because every machine booted into a BASIC interpreter which was stored as a ROM on the motherboard. There were other languages for the C64 including assembly language, Fortran, C and Pascal, but the most readily accessible was BASIC. Let's learn how to use BASIC on the C64.

We start with a classic, Hello World! It's a useful tool to learn the syntax and structure of BASIC. Type the following and press ENTER at the end of each line:

```
10 PRINT "HELLO WORLD!"
20 GOTO 10
```

These two lines start with a number, 10 and 20. This is our line number, just like other languages. These line numbers help us to identify any errors by directing us to a specific line. We use 10, 20, 30... in BASIC to give us the space to insert lines of code between them should we need to fix an issue or add a feature.

Line 10 will print a sentence to the screen, in this case "HELLO WORLD!". Line 20 is GOTO and this is an instruction that the code needs to go back to line 10 and repeat the process. In Python the closest example would be a while True loop that will continuously run the code within it. To run this code example type RUN and press ENTER. You'll see "HELLO WORLD!" printed continuously to the screen. Press ESC (RUN/STOP) to break the loop and stop the code.

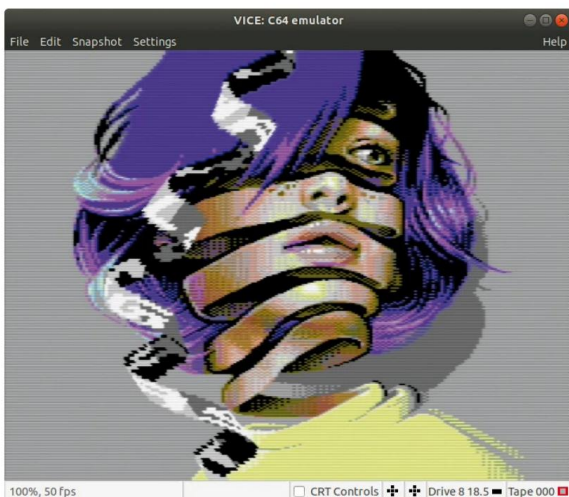
Sometimes we only need a loop to iterate (go round) for a set number of times and for this we need to use a for loop. Here is the code.

```
10 FOR X=1 TO 10
20 PRINT "LXF ROOLZ"
30 NEXT X
```

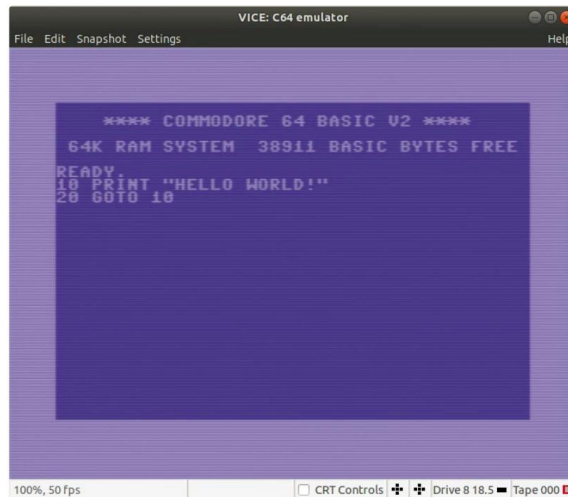
Line 10 creates a variable "X" which stores the current loop counter number which starts at 1 and increases by 1 each time the loop iterates. When the loop counter reaches the TO value of 10, the code exits. Line 20 reuses the PRINT statement to print a message on-screen. Line 30 will iterate the value of X, the loop counter, each time the loop iterates. Type RUN to see the code in action.

## A BASIC app

The best way to learn anything is by doing, so let's make a classic game in code. Take the Magic 8-ball: we think of a question, and then shake the 8-ball. An answer is



C64 demos are remarkable – they push the hardware to its limits. This demo is Listen With Your Eyes by Boozie Design, created in 2020!



Everyone has created a Hello World project in a language, and this is how we do it in C64 BASIC.

revealed in the murky liquid that answers our question. We'll code our own version in BASIC.

We start by giving an instruction to the user via a PRINT statement (line 10).

```
10 PRINT "ENTER FIVE RESPONSES"
```

Using a FOR loop that iterates five times (line 20) we print an instruction to the user and then capture the user INPUT to an array called S, using the current value of I as a numerical position (index) in the array (line 30). Then we iterate the value of I to repeat the FOR loop until we have five responses (line 40).

```
20 FOR I=1 TO 5
30 PRINT "RESPONSE";I;:INPUT S$(I)
40 NEXT I
```

Lines 50 and 60 print an instruction to the user. Pressing Y or X will see the input captured and stored in a variable set to store string values A\$.

```
50 PRINT "THINK OF A QUESTION AND PRESS Y AND ENTER TO START"
60 INPUT "PRESS Y TO START OR X TO EXIT"; A$
```

Line 70 sees the start of a conditional statement, which checks the value stored in the variable A\$. If the value stored is Y then the code will skip forward (GOTO) line 100. Else it just carries on.

```
70 IF A$="Y" THEN GOTO 100
```

## QUICK TIP

To see all of the code in a project we need to use LIST. This will print the code line by line. We can also specify a line number to start at, for example, LIST 100 and it can be used to list the contents of a floppy disk.

## » HISTORY OF BASIC

The BASIC language has been with us since 1964 and it was developed as a high-level (easy to read) language by John G Kemeny, chairman of the Dartmouth College Math Department. The name BASIC came from Thomas Kurtz. The structure of BASIC is meant to be oversimplified on purpose. Each line of BASIC is read by the compiler and turned into byte code that the machine can then run.

BASIC can be compiled or interpreted, depending on the version which you are using. On the Commodore 64 we see Commodore 64 BASIC V2, a version of Microsoft BASIC, which has a limited number of keywords, but other versions of BASIC exist for the C64 and the most popular alternative choice was Simons' BASIC, which was written in 1983 by David Simons when he was 16 years old. Simons' BASIC had extra keywords, 114 in total, for graphics, sprites and disk operation. Magazines of the era praised Simons' BASIC, stating that it turned the Commodore 64 into a brand new computer. Simon' BASIC is available on floppy disk and cartridge via online auctions and usually sells for quite a high price given its coveted status in the community.





## QUICK TIP

The floppy image created in this article can be used with any emulators which run .d64 image files. If you have a real Commodore 64 and an SD2IEC floppy drive emulator, that image can also be run on a real machine!

Line 80 assumes that the user pressed X and wants to end the game. It prints a goodbye message before moving to line 90 that instructs the code to go to the last line of code at line 130.

```
80 PRINT "THANKS FOR PLAYING"
90 GOTO 130
```

Line 100 is where we rejoin the sequence if the user chose to play the game. We create a variable, X, and in there we store a random integer between 1 and 5.

```
100 X=INT(RND(1)*5)
```

Line 110 prints a value stored inside the array, S, created at line 30. The value returned from the array is chosen using the random number stored in X.

```
110 PRINT S$(X)
```

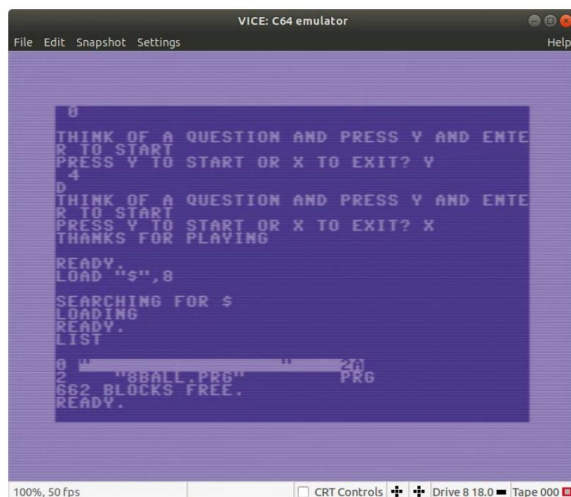
Line 120 is only activated if the user chose to play the game, and it directs the player back to make that decision again. If so the code will loop once again.

```
120 GOTO 50
```

The final line is 130 and here the code ends. We only reach this line if we choose to end the game at line 60.

```
130 END
```

Run the code by typing RUN and follow the on-screen instructions. After running the code a few times, press X



LIST is the BASIC way of showing the contents of a disk or a program that's been loaded into memory.

## » USING THE GPIO

If we mention GPIO then our natural response is "Raspberry Pi" but the Commodore 64 also has a GPIO and it's called the User Port. Located on the rear of the machine. The user port provides eight general and output pins and these can be controlled using BASIC commands. By "poking" (inserting a value into a specific memory address) we can insert instructions which will turn the "pins" of the user port into inputs/outputs and set their default value. To use the user port in this way you'll need a breakout board because there are exposed pins on the board that, if used incorrectly, can damage your Commodore 64. You can make your own breakout board with a connector or you can buy a ready-made board from one of the many community members who sell on Tindie.

The user accessible GPIO pins run at 5V, which offers the chance to connect to an Arduino or other 5V microcontroller, but not a Raspberry Pi because it runs a 3.3V. Flashing an LED should be your first project and this blog post explores the process in greater detail: <https://bigl.es/friday-fun-making-an-led-blink-with-a-commodore-64>. From this post you can boost your knowledge and create more projects including controlling motors and building robots!

and Enter to exit back to the interpreter. Now that we have written some code, we need to save it. For this click File>Create and attach an empty disk and select Unit 8. Name the disk **LXF.d64**. Now go back to File>Attach Disk Image>Unit 8 and select **LXF.d64**. This will insert a virtual floppy disk into drive 8, the default drive number for the original 1541 floppy drive.

In the emulator type the following to save your work to disk.

```
SAVE "8BALL.PRG",8,1
```

After a few seconds your code is safely stored on the disk. Next, to see if the code has been saved correctly, type the following. Remember to press ENTER at the end of each line.

```
LOAD "$",8
```

```
LIST
```

You should see **8BALL.PRG** listed. To load the game type the following:

```
LOAD "8*",8
```

To run the game, first we need to check that all of the code is present and then we can test the game.

```
LIST
```

```
RUN
```

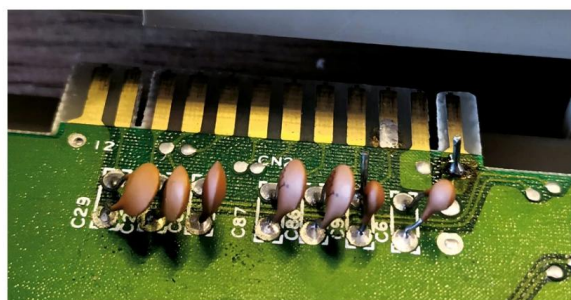
In this short tutorial we've learnt to write BASIC code using core coding concepts: loops, conditionals, variables, data types and user input. We've also learnt how to create blank floppy disks, and how to save and retrieve our code from them.

## Demo scene

The Commodore 64 has a thriving community and demo scene, even after all of this time. In case you didn't already know, the demo scene is where bedroom coders developed animations, music and elaborate music videos that pushed the power of the Commodore 64 to the limit.

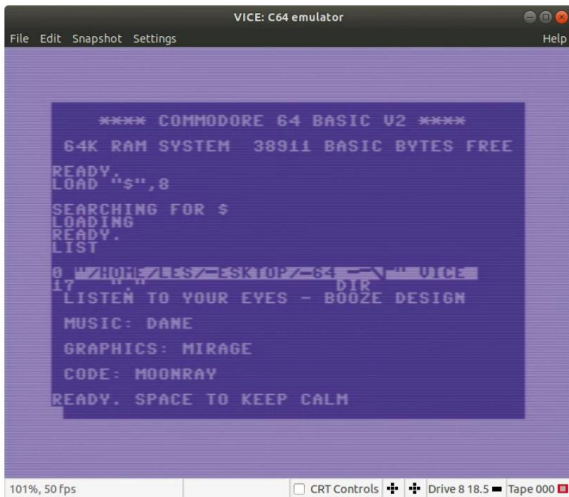
Demo scene coders often used C or assembly language to write their code because it provided the fastest access to the resources of the machine. Assembly language also benefited from pushing the Commodore 64 even further than the creators had thought possible, including rudimentary 3D graphics and image manipulation.

The biggest selling point of the Commodore 64 was its ability to create music. The SID (Sound Interface Device) chip, officially known as the MOS 6581/8580, was an extraordinary sound chip for its time. It featured three separately programmable independent audio oscillators and four different waveforms per audio oscillator, and with it clever coders could produce astounding music. To this day musicians prize this chip and seek it out for use in their own electronics



The User Port is the GPIO of the Commodore 64 and can be used with electronics to create hardware projects.





Loading a file from disk means that we can run games or applications from either disk or tape.

instruments. Even in 2020 there are still demos that push the hardware to its very limits, and considering that the Commodore 64 is 37 years old, they achieve remarkable results.

The demo scene sprang from a more “dark” place, namely the copying groups of the 1980s. These were groups who competed to get the latest games and share them over Bulletin Board Service or at copying parties. Each group wanted to show their prowess at breaching the various copy-protection tools. They created a series of demos that were included at the start of a game to identify the team which “cracked” the copy protection and added other features, such as game trainers which provided infinite lives, time, bullets and so on.

## Download your demo

These demos are now saved for posterity in many different Commodore 64 tribute sites and that provides us with ready access to them. We went to <https://csdb.dk> and looked for the latest demo. We found one, Listen To Your Eyes, which was created 28 June 2020. After clicking the download link we had a .prg file. But how can we use it? Luckily, Vice can work with files on our computer and to set this up we need to take the following steps.

Create a new blank directory called DEMOS on your Desktop. Next, download the demo (<https://csdb.dk/release/?id=192916>) and copy it to that directory.

The demo file can't be opened like a disk, but if we go to Settings>Drive Settings>Drive #8 Options and Enable IEC device, then click File System Directory... and navigate to the DEMOS directory on your desktop.

In Vice, type the following command to load and list the contents of that directory:

```
LOAD "$",8
```

```
LIST
```

To load the demo.

```
LOAD "LISTEN TO YOUR EYES.PRG",8,1
```

```
RUN
```

The demo will take a little time to load, even on a fast computer, because we're emulating the speed of an

original C64. If you wish to speed up the emulator go to Settings>Maximum Speed.

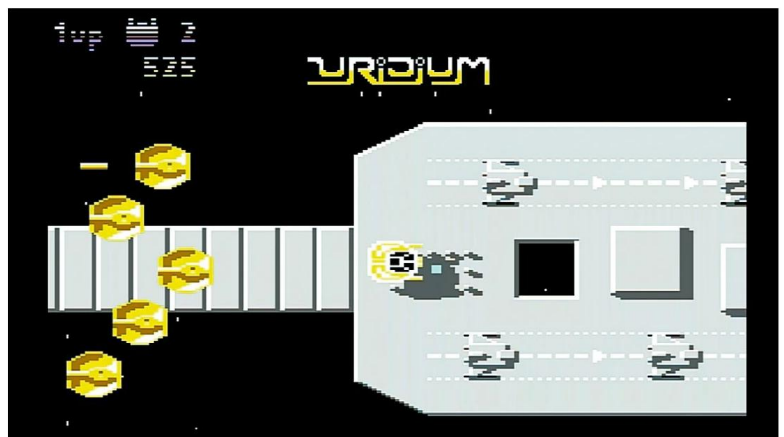
## Commodore community

Despite its age the Commodore 64 has seen a resurgence in recent years with a community that feels just like the Linux community. Community members are musicians, coders and retro gamers who exploit the C64 to the max. Sites such as [www.lemon64.com](http://www.lemon64.com) and forums such as <https://everythingc64.boards.net> offer places to talk with fellow enthusiasts and share projects and ideas.

The cost of second-hand machines can vary from £50 to hundreds for a low-value serial number model. Collectors are seeking kits to revive the old machines and businesses such as TFW8B and sellers on sites such as Tindie are offering kits to modify your C64. Adding a newer kernel, replacing the capacitors and keyboard repairs are just some of the simple kits on offer. There are also kits on sale that can be used to get the C64 online to a BBS via an ESP32 microcontroller and the user port located on the rear of the C64. Sven Petersen has a range of diagnostic tools and adapters to ensure that our Commodore 64 is protected and ready for the next 40 years. His C64 Saver PCB is a must-buy for second-hand C64 as the original power supplies can go bad and damage your prized machine. Sven's GitHub repository (<https://github.com/svenpetersen1965>) is a gold mine of resources and information.

It would be remiss of us not to talk about the ongoing legacy of the Commodore 64. It was and still is a community of great individuals who repair and resurrect old machines. Write blog posts and record how-to videos showing projects created using the legendary machine.

With emulation we can scratch the surface of this great machine, we can learn how it works and what made it great, but there's nothing like having the real hardware. Turning on the machine for the first time and writing a few lines of BASIC is nostalgic, but also calming. We have no distractions, no IDE or framework. Just access to the bare machine, which we can immediately communicate with using just a few lines of code. Mindfulness on an early piece of home computing – who would have thought it? **LXF**



Uridium is a classic shoot-em-up that's still playable to this day, but are you quick enough to beat the aliens? Only one way to find out...

» GET MORE NOSTALGIC EMULATION Subscribe now at <http://bit.ly/LinuxFormat>



Credit: <https://godotengine.org>

GODOT

**Part Two!**  
Missed part  
one? Get hold of  
it by turning  
to page 69

# Create a video game in the Godot engine

Following on from last issue's introduction to the development engine Godot, **Calvin Robinson** shows you how to develop your own game.



**OUR  
EXPERT**

**Calvin Robinson** is a subject matter expert at the National Centre for Computing Education, and a computer science teacher.

**G**odot is an open source video game development engine. In last issue we looked at the engine platform and produced our first application: a 2D interface that displayed Hello World. This issue we're going to use Godot to develop our first video game.

We're going to start off with a dodge 'em game, reminiscent of *Asteroids*, with enemies swarming a two-dimensional environment and the player having to avoid them. However, we won't have a weapon to begin with, so to score any points we'll have to survive for the longest amount of time possible, without colliding with any enemies.

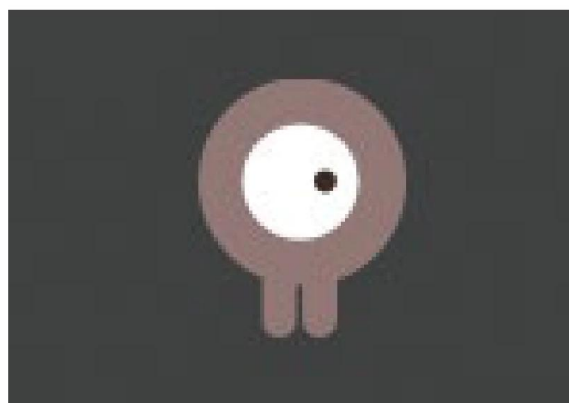
Two-dimensional games are simpler to create, so we'll start there, before jumping in to a three-dimensional environment.

Let's begin with a brand new project in Godot. Assuming that you followed along from last issue, launch Godot. Otherwise, make sure you have everything installed.

## Full Steam for Godot

The easiest way of getting Godot downloaded, installed and kept up to date is by using Steam. A simple `sudo apt-get update && sudo apt-get install steam` should get Steam sorted on a Debian-based distro. There's also a .deb installer available at [SteamPowered.com](https://www.steampowered.com).

Run Steam for the first time, accept the terms and conditions and then install the font packages. Sign up and sign in, and you're ready to go.



■ Spawning an alien as our player character.

On the Steam store page, search for Godot, install it for free and it'll appear in your library. Depending on your Steam settings, Godot should now stay up to date at all times. Use the library to launch Godot. You can add a desktop and/or start menu shortcut by right-clicking and going to Properties.

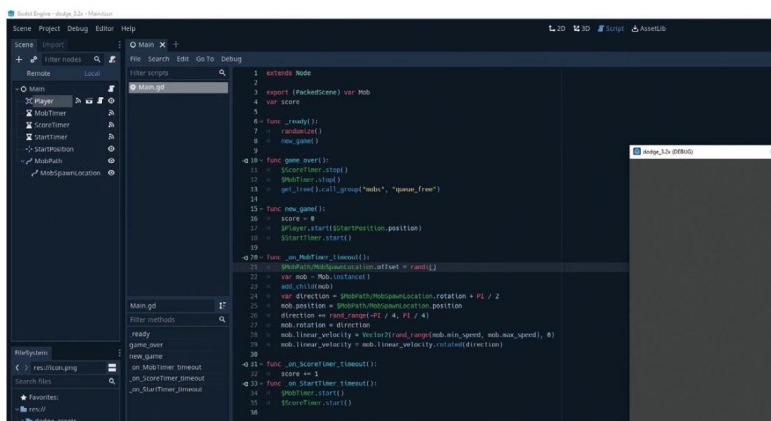
In the Godot Project Manager create a new Project, setting the window resolution to 480x720 by going to Project Settings>Display, Window. We'll also want to set the Stretch option to 2D Mode and tick Keep on the aspect ratio to ensure nothing goes off-kilt and the aspect ratio remains in place at all times. We're developing a game in portrait mode, and we don't want things to be shifted too much at any point.

We'll need the dodge assets from Godot's wiki page (<https://docs.godotengine.org>): download them from <https://bit.ly/38RFPgS>. This package includes all the sprites and sound effects we'll need, so that we can focus on building the game itself, rather than spending time creating such assets. Of course, if you want to edit these or create your own assets from scratch to give your version of the game a personal touch to it, feel free to do so. Everything is open source.

## Scenes and Nodes

We've discussed the setting up of Scenes and Nodes, but as a brief recap; a Scene is an environment of sorts, or a collection of Nodes, whether visible or not. Scenes can be tested independently of each other. Nodes are

Godot editing suite is incredibly intuitive by design.





classes: objects with properties that we'll use in our game. We'll need three Scenes for this project: Player, HUD and Mob. We'll group all three Scenes into a parent Scene called Main. It might also be a good idea to save these Scenes into a folder to keep things organised, but also as it's considered best practice. Godot has a great filing system that stores all assets locally, relative to each other.

Starting with the Player scene, let's start adding some objects. Godot will require us to specify a type of Node, so we'll go with 'Other' in this instance, even though we're creating a two-dimensional environment. Other enables us to use the search – select Area2D from the auto-complete list. Ignore any warnings we get at this point.

Area2D includes some basic collision detection, so we'll know if any other object is interacting with our player character. Rename the Node by right-clicking or double-clicking, and call it Player. This is now our primary Node in the Player scene. We can add child Nodes to it later, if necessary, which can inherit the properties of the parent Node. This is a typical object-oriented design approach.

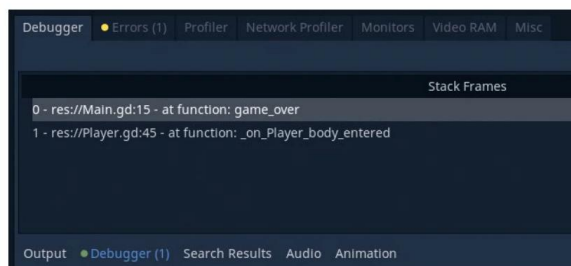
Our first child Node will be AnimatedSprite, to control our player character's animations. Right-click Player to add a child, then rename it as such. Click Frames (a property in the Inspector) and select New SpriteFrames followed by SpriteFrames, to start a new animation. Select Default and rename it to Walk. This will be our first animation.

## Moving the alien

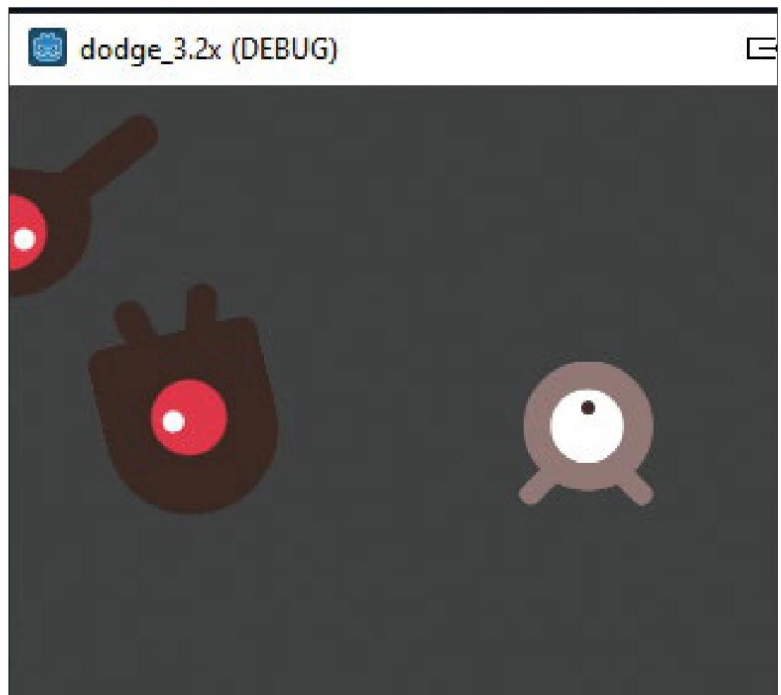
Now let's add additional animations for directions. Press New Animation and create one called Up. Switch to the FileSystem tab and you should see images of playerGrey\_up and playerGrey\_walk; drag and drop these images onto our new animations. You'll see a funky little alien guy with an enlarged eyeball; one static, one with feet apart as if it's walking. Scale these down by clicking them and then setting the scale to half size: that's 0.5 on both the x- and y-axis in the Inspector's Node2D settings.

Now we're familiar with Nodes, let's create another child of Player called CollisionShape2D. This will be our hit box that determines the area around our player character that's measured for collisions. Find 'Shape' in the Inspector menu and select New CapsuleShape2D, then drag the shape to until it's the appropriate size for our alien sprite. Save the Scene. That's our Player set up and almost ready to be tested.

Before we do begin testing, it's worth adding some functionality to our Player Scene. Time for our first script. Select Player and press Attach Script, then



The Debug window is incredibly helpful for spotting errors.



Create. Godot accepts GDScript, a proprietary scripting language that's not too dissimilar to Java, or C#.

```
extends Area2D
export var speed = 400 # How fast the player will move (pixels/sec).
var screen_size # Size of the game window.
```

Here you'll notice we're inheriting from Area2D, naturally, and exporting a variable called `speed`. This sets the appropriate value in Inspector to match, in this case, 400. Exporting values in this way enables us to adjust the properties of our Nodes (objects) via code.

```
func _ready():
    screen_size = get_viewport_rect().size
```

Here we've set up a new function, labelled `_Ready()`, which will run every time our Node is spawned within this Scene. For that reason, we're grabbing the window resolution, so we have it up front.

We'll also need a function to check for user input, move the sprite accordingly, and display the animations while it does so. For this we'll create a second function called `_Process()`:

```
func _process(delta):
    var velocity = Vector2() # The player's movement vector.
    if Input.is_action_pressed("ui_right"):
        velocity.x += 1
    if Input.is_action_pressed("ui_left"):
        velocity.x -= 1
    if Input.is_action_pressed("ui_down"):
        velocity.y += 1
    if Input.is_action_pressed("ui_up"):
        velocity.y -= 1
    if velocity.length() > 0:
        velocity = velocity.normalized() * speed
        $AnimatedSprite.play()
    else:
        $AnimatedSprite.stop()
```

This function monitors key presses and increases velocity on the appropriate x- or y-coordinates, where

Our little alien learns to swim.

## QUICK TIP

If Godot displays "Attempt to call function 'play' in base 'null instance' on a null instance" then this is a syntax error. It means there's probably a typing mistake in the name of one of the nodes. Godot is case-sensitive.





## QUICK TIP

During collision detection, errors can occur because of the timing of events. Using `set_deferred()` reminds Godot to wait until it's safe to disable the appropriate sprite.

positive number being for right or up, and negative numbers being for down or left. If any movement takes place, the animation sprite is played. When movement stops, so does the animation.

## Movin' and groovin'

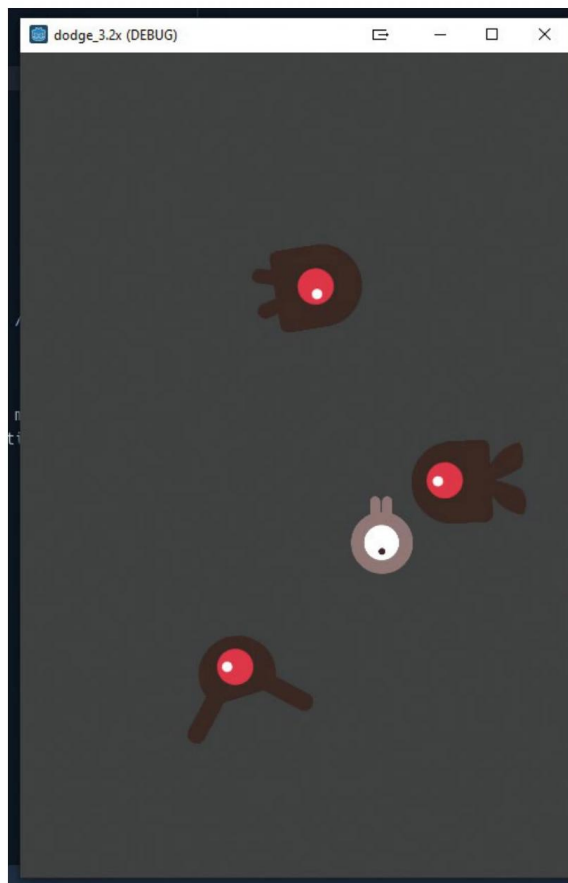
We have movement! However, we've not set anything in place to prevent our player character from walking out of bounds – beyond the limits of our game environment. We'll need to use a function called `clamp()` to prevent this. At the bottom of our `_Process()` function we'll need to make some additions:

```
position += velocity * delta
position.x = clamp(position.x, 0, screen_size.x)
position.y = clamp(position.y, 0, screen_size.y)
```

Okay, now we've got our sprites created and drawn, and we're monitoring keyboard presses to move our player character. Now we're ready to begin testing. Hit F6 to save and launch. We should be able to move our alien around the screen, but not outside of it.

So our player moves around the game environment, but it doesn't look right. While the animation does play, it doesn't yet know which way it's facing. We'll need to mirror the animation if we move our character in an opposite direction. Let's update our `_Process()` function again:

```
if velocity.x != 0:
    $AnimatedSprite.animation = "walk"
    $AnimatedSprite.flip_v = false
    # See the note below about boolean assignment
    $AnimatedSprite.flip_h = velocity.x < 0
elif velocity.y != 0:
    $AnimatedSprite.animation = "up"
    $AnimatedSprite.flip_v = velocity.y > 0
```



We've created  
a monster!

In our `_Ready()` function we'll need to add the `hide()` if we're using GDScript or `Hide()`; for C#, to make the player character hidden on game launch.

If we run out the game again (F6) our player should appear on keypress and the animations should properly reflect the direction of travel.

Our player detects the boundaries of the game world, but doesn't yet know if he's hit an enemy, because we haven't created any. Let's prepare our alien for collisions by adding a signal alert right under `extends Area2D`. In GDScript it's a simple `signal hit`. In C# it's:

```
[Signal]
public delegate void Hit();
```

Using `signal hit`, our game picks up an alert every time our character node collides with another node, or is 'hit'. To finish setting this up, select the Player node and switch from Inspector to the Node tab. We should now see a `hit()` signal. Under `Area2D` select `body_entered(body: Node)` so that we can receive alerts when the body area of our alien has been hit. Then click Connect followed by Connect a Signal, and Connect one more time. The signal should now be set up. We just need to add the code to our function:

```
func _on_Player_body_entered(body):
    hide()
    emit_signal("hit")
    $CollisionShape2D.set_deferred("disabled", true)
```

Godot will be signalled every time something hits our alien and the sprite will disappear, thus killing our player. We've set things up so that we disable collision detection once it's occurred, so that we don't spam the process.

Since we've killed off our player character, we'll need to insert some code to restart it on a new game:

```
func start(pos):
    position = pos
    show()
    $CollisionShape2D.disabled = false
```

Let's get that enemy added to our game. We'll need a new Scene (click Scene>New Scene) and call it Mob, short for mobile – a computer controller character. We'll have a swarm of mobs in our game at any given time. Select the `RigidBody2D` variety of Node for our mob. Add a few children Nodes: `AnimatedSprite`, `CollisionShape2D` and `VisibilityNotifier2D`. Gravity can be set to zero in the `RigidBody2D` properties, using the Inspector tab. We might also want to select Mask in the `PhysicsBody2D` property box, to make sure mobs can't collide with each other. We only want them taking out our player, not themselves!

`AnimatedSprite` will need graphical animations setting up, just as we did earlier for our player character. Import the images for swim, fly and walk. Setting the FPS (frames per second) alters the speed of each animation. Be sure to enable Playing to On, in the Inspector window. Scale the images down in size again, and add a `CapsuleShape2D` for collision, again, like we did with our player character. We may need to use Rotation Degrees to alter the direction of the image. One satisfied, save the Scene.

We will, of course, need to add a script to our Mob, setting the minimum and maximum speed ranges:

```
extends RigidBody2D
export var min_speed = 150
export var max_speed = 250
```



Of course, we'll need a `_Ready()` function, similar to an `init()` in other languages, to declare what happens on launch:

```
func _ready():
    var mob_types = $AnimatedSprite.frames.get_
    animation_names()
    $AnimatedSprite.animation = mob_types[randi() %
    mob_types.size()]
```

Putting the different animations into a list, such as ["fly", "walk", "swim"] we can use said list as a series of frames by selecting a number from 0 to 2 and using a different one for each mob we spawn. The `randomize()` function enables us to acquire a different sequence of numbers every time we execute the scene. We'll implement this in our Main scene, shortly. One last thing on the enemies though: unlike our player character we want our enemies to whiz out of the game environment. We just need to ensure they die once they head out off the screen:

```
func on_VisibilityNotifier2D_screen_exited():
    queue_free()
```

That's nearly all of our Scenes set up. The only thing left to do is build our Main Scene, which will bring them all together. Set up a new Scene and Node called Main, then create a new instance of our Player by clicking Instance and selecting **Player.tscn**. This is essentially like spawning a new object from the same class.

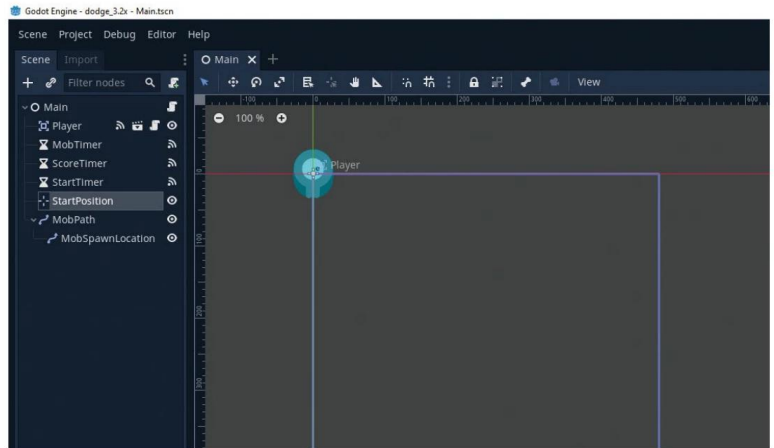
Add a **Position2D** Node as a child to Main and call it **StartPosition**. Then add three Timer Nodes and call them **MobTimer**, **ScoreTimer** and **StartTimer**, respectively. These timers will control how often mobs spawn, increase the score for every second our player stays alive, and add a delay before starting a new round. The **StartPosition** is literally a starting point, to present where we'd like our player character to spawn on a new game. Alter the properties of these timers, setting **MobTimer** to a half, **ScoreTimer** to 1 and **StartTimer** to 2. **StartTimer** also has a property called One Shot, so enable that. Set the **StartPosition** coordinates to x=240 and y=450.

## Send in the Mobs

Mobs will need to spawn in random locations, preferably along the outer rim of the screen. Add a new **Path2D** Node called **MobPath**, as a child to Main, using the Add pointer (with the green plus sign) to draw a path. Click anywhere in the game world to add points to the path. In this case it's probably easier to just click roughly in each four corners of the game world. Once we're happy with the path, click the Lock Path button (a magnet icon). It's best to draw the path in a clockwise fashion, otherwise we'll have to mirror the animations we set up earlier, because our mobs will be facing the wrong direction. Add a **PathFollow2D** child Node to **MobPath**, called **MobSpawnLocation**, so that we can use it as a random positioner to direct mobs along the path.

Finally, we need to add a script to our Main scene:

```
extends Node
export (PackedScene) var Mob
var score
func _ready():
    randomize()
```



Clicking the Main Node should reveal Mob properties. Click the [Empty] drop-down and select Load, then choose **Mob.tscn**. Do the same for Player and check the signals. We should see 'hit', which we set up earlier. Click Connect and choose `game_over`. We'll need to add some code to implement a `game_over` function, as well as a `new_game` function to restart the game after it ends.

```
func game_over():
    $ScoreTimer.stop()
    $MobTimer.stop()
func new_game():
    score = 0
    $Player.start($StartPosition.position)
    $StartTimer.start()
```

There should also be a `timeout()` signal for each Timer that we set up. Connect those to the main script, too. We'll need a couple of small functions:

```
func on_StartTimer_timeout():
    $MobTimer.start()
    $ScoreTimer.start()
func on_ScoreTimer_timeout():
    score += 1
```

Finally, we'll need to set up a function to randomly select spawn points for our mobs. We'll choose a random location with **Path2D**, create an instance of our mob, set it moving in a clockwise direction and then set its speed:

```
func on_MobTimer_timeout():
    $MobPath/MobSpawnLocation.offset = randi()
    var mob = Mob.instance()
    add_child(mob)
    var direction = $MobPath/MobSpawnLocation.
    rotation + PI / 2
    mob.position = $MobPath/MobSpawnLocation.
    position
    direction += rand_range(-PI / 4, PI / 4)
    mob.rotation = direction
    mob.linear_velocity = Vector2(rand_range(mob.min_
    speed, mob.max_speed), 0)
    mob.linear_velocity = mob.linear_velocity.
    rotated(direction)
```

Pop `new_game()` into our `_Ready()` function (or `NewGame()`; if you're using C#) and that's the lot! Save and load with F6. We should now have a fully functioning dodge game, created entirely in Godot. **LXF**

Everything is coming together nicely. Scenes, assets and Nodes.

### QUICK TIP

To ensure the Player and Mob sprites can't be accidentally moved or resized while we're working in Godot, set "Make sure the object's children are not selectable" in node settings to lock them in place.

» **STOP PLAYING WITH OUR ♥s AND** Subscribe now at <http://bit.ly/LinuxFormat>



# Recreate the Game of Life simulator

Calvin Robinson uses Python and Turtle graphics to run this unique game.



**OUR  
EXPERT**

**Calvin Robinson** is a subject matter expert at the National Centre for Computing Education, and a computer science teacher.

**E**ach issue we're building a retro video game. In this series of programming games with Python, so far we've created a lunar space module, a side scrolling platformer, a 2D graphical insect/animal game, *Pac-Man* and *Pong*. Take a look at the **Linux Format** back catalogue to have a go at creating any of these vintage game formats (*more details on page 69*).

This issue we're going to be creating life itself. A simulation of life, that is. *Game of Life*, sometimes known as just *Life*, is a unique game in that it doesn't have any interactive properties, per se. The game is set up or configured and then left to run, simulating the creation of life exponentially. Bear with us on this one...

John Horton Conway, a British mathematician, invented the *Game of Life* in 1970 based on the work of John von Neumann in 1940. The very same John von Neumann, a Hungarian mathematician, who would later go on to design von Neumann architecture – the basis of all modern central processing units (CPUs). We've no idea why all 20th century mathematicians were apparently called John. Von Neumann's original theory was that a being or organism should be able to reproduce itself and pass the tests of the Turing machine, in order to call itself 'life'.

*Game of Life* simulates an unpredictable environment and monitors the behaviour of cell automata based on a set of predefined rules. This is similar to the principle still used in contemporary virus behaviour modelling. No doubt mathematicians around the world have been using advanced adaptations of this methodology to predict Coronavirus spread rates. To prove that the automaton was 'alive' the game would have to be able to run for a long period without the cells dying, and the original configuration would need run indefinitely, without any loop cycles.

### The rules of life...

- There should be no explosive growth.
- There should exist small initial patterns with chaotic, unpredictable outcomes.
- There should be potential for von Neumann universal constructors.
- The rules should be as simple as possible, while adhering to the above constraints.

A 2D game world is required, in which every pixel is binary and therefore either dead or alive. Pixels, known as cells, interact with the cells on all four sides and all four diagonal corners.

### The game rules:

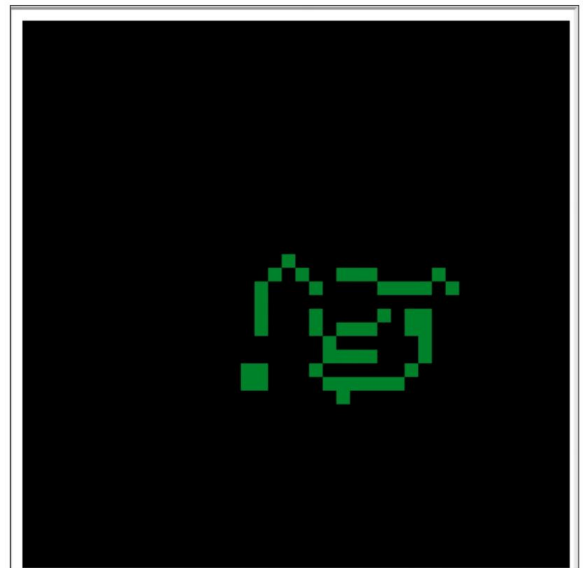
- Any live cell with fewer than two live neighbours dies, as if by underpopulation.
- Any live cell with two or three live neighbours lives on to the next generation.
- Any live cell with more than three live neighbours dies, as if by overpopulation.
- Any dead cell with exactly three live neighbours becomes a live cell, as if by reproduction.

To compare the behaviour of each living automaton to that of a real life creation, the following rules are considered:

- Any live cell with two or three live neighbours survives.
- Any dead cell with three live neighbours becomes a live cell.
- All other live cells die in the next generation. Similarly, all other dead cells stay dead.

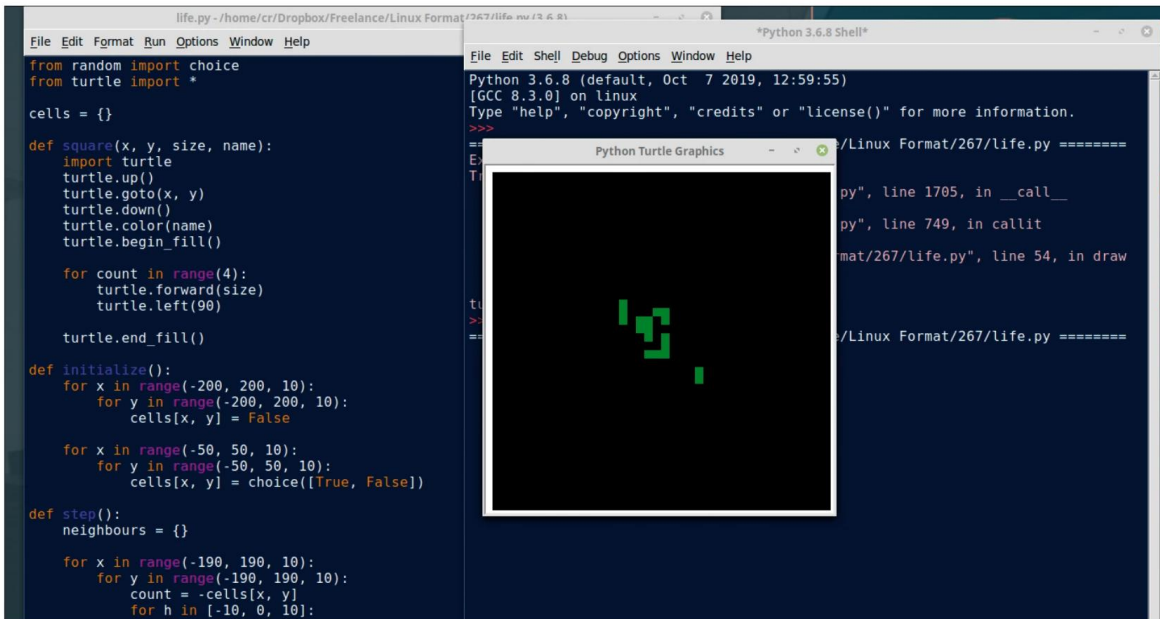
### QUICK TIP

If you want to take things to the next level, Golly is an open-source program for simulating cellular automata at an incredible rate, sticking to Conway's Game of Life principles.



In Game of Life, green automata are generated with no user input.





Turtle graphics are used to visualise the game's cells.

All of these rules are considered when setting up the initial 'seed' and must be applied to every single cell at spawn. Cells can be born at the same time that others die, each lifecycle is referred to as a tick.

### Coding our zero player game.

We'll need to import a few modules before we get going. As ever, we don't like to start completely from scratch if we can help it. Python already includes a great class for 'choice' within random. Turtle graphics will be needed to draw our cell shapes on the screen. We'll include a working version of the full source code on our website in the Archives section, as well as a copy on the disc image and **LXF** website archive for this issue.

```
from random import choice
from turtle import *
```

We'll also want to declare one primary variable, cells, as an empty list:

```
cells = {}
```

The code to get our *Game of Life* working is relatively simple. We'll need to set up four methods: one for creating our squares in Turtle graphics, a method for initialising our cells; another for computing each step or tick of the game; and finally a method for drawing the squares we drew:

```
def square(x, y, size, name):
    turtle.up()
    turtle.goto(x, y)
    turtle.down()
    turtle.color(name)
    turtle.begin_fill()
    for count in range(4):
        turtle.forward(size)
        turtle.left(90)
    turtle.end_fill()
```

Given an x- and y-coordinate, followed by a length for the size and a name, this method will create a square ready to be drawn in-game.

Once our squares have been created, we'll need to draw them:

```
def draw():
    step()
    clear()
```

```
for (x, y), alive in cells.items():
    color = 'green' if alive else 'black'
    square(x, y, 10, color)
update()
ontimer(draw, 100)
```

To randomly initialise our cells:

```
def initialise():
    for x in range(-200, 200, 10):
        for y in range(-200, 200, 10):
            cells[x, y] = False
```

### QUICK TIP

Cellular Automaton Sequencers including Game Of Life Music Sequencer for iOS and glitchDS for the Nintendo DS use the Game of Life to produce music as MIDI sequencers.

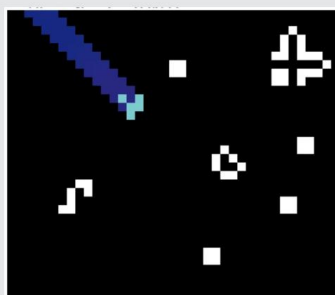
## » LIFE OUT THERE

Conway's *Game of Life* has a worldwide community of enthusiasts intent on constructing new types of cell structures. 2020 has led to some interesting discoveries in the area of stable circuitry, including some new glider guns and spaceships. A spaceship is a construct that travels across the environment, whereas a glider gun is a construct that's stationary, while producing gliders or other spaceships that spread from it. Essentially, a gun fires out spaceships as ammunition. Gliders are the lightest, smallest type of spaceship.

Gliders can interact with other structures in all kinds of ways, including destroying them. The really interesting prospect though, is that gliders can be used to construct logic gates, in the typical AND, OR and NOT states. This explains why *Game of Life* is Turing complete, because it's theoretically possible to simulate a computer with infinite time and memory. Sometimes guns can fire gliders in such a way that they construct entirely new objects.

This year saw the exciting discovery of the first guns that can fire c/5 orthogonal spaceships and c/5 diagonal spaceships, as well as a

colour-changing glider. More information can be found at [www.conwaylife.com](http://www.conwaylife.com), where new posts and user forums share recent discoveries on a regular basis.



A simple glider spaceship collides with an object and explodes.



## QUICK TIP

Game of Life is Turing Complete, which means it can simulate a universal Turing machine and/or compute every Turing-computable function.

```
for x in range(-50, 50, 10):
    for y in range(-50, 50, 10):
        cells[x, y] = choice([True, False])

    For each tick in the Game of Life, we compute a step
    with our method:
    def step():
        neighbours = {}

        for x in range(-190, 190, 10):
            for y in range(-190, 190, 10):
                count = -cells[x, y]
                for h in [-10, 0, 10]:
                    for v in [-10, 0, 10]:
                        count += cells[x+h, y+v]
                neighbours[x, y] = count

        for cell, count in neighbours.items():
            if cells[cell]:
                if count < 2 or count > 3:
                    cells[cell] = False
            elif count == 3:
                cells[cell] = True
```

We've set up a new list to track neighbouring cells. Birth of new cells will happen with three neighbours, and death will occur with fewer than two or greater than three neighbours. Finally, set everything up and run the following methods:

```
setup(420, 420, 370, 0)
hideturtle()
tracer(False)
initialise()
draw()
done()
```

Now we can run our game and the automata should begin replicating. Based on our initialisation and step methods, the cells will check the cells around them (see `neighbours`) and see if they're occupied or vacant, before deciding to move into them as the automata attempts to self-replicate. The longer we leave the game running, the more interesting it can become.

## Generations of life

As the game progresses we see what we'll call 'generations' of life, and with them different recognisable patterns. There are many common changeless cell structures, we call 'still lifes', which will remain in their shape regardless. Then there are the 'oscillators', which return to their original form after a number of generations. 'Spaceships' is what we call cell structures that traverse across the game world. What's interesting, is that there are certain shapes of still lifes, oscillators and spaceships that will commonly appear, regardless of the starting configuration.

The finished code should look something like this:

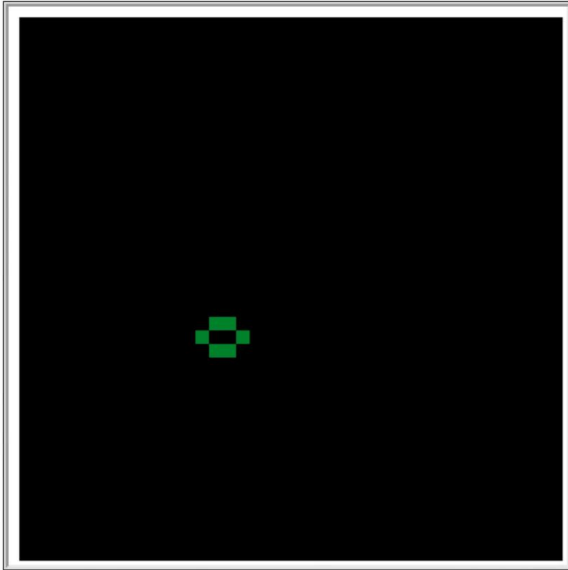
```
from random import choice
```

Still lifes		Oscillators		Spaceships	
Block		Blinker (period 2)		Glider	
Bee-hive		Toad (period 2)		Light-weight spaceship (LWSS)	
Loaf		Beacon (period 2)		Middle-weight spaceship (MWSS)	
Boat		Pulsar (period 3)		Heavy-weight spaceship (HWSS)	
Tub		Penta-decathlon (period 15)			

Still lifes, oscillators and spaceships make up the common cell structures.

Credit: [https://en.wikipedia.org/wiki/Conway's\\_Game\\_of\\_Life#Examples\\_of\\_patterns](https://en.wikipedia.org/wiki/Conway's_Game_of_Life#Examples_of_patterns)





A still life, with no more creation. This particular game of life was over before it even began.

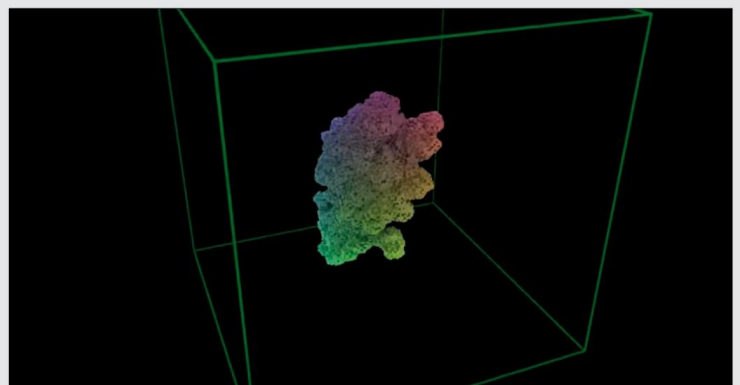
```
from turtle import *
cells = {}
def square(x, y, size, name):
    import turtle
    turtle.up()
    turtle.goto(x, y)
    turtle.down()
    turtle.color(name)
    turtle.begin_fill()
    for count in range(4):
        turtle.forward(size)
        turtle.left(90)
    turtle.end_fill()
def initialise():
    for x in range(-200, 200, 10):
        for y in range(-200, 200, 10):
            cells[x, y] = False
    for x in range(-50, 50, 10):
        for y in range(-50, 50, 10):
            cells[x, y] = choice([True, False])
def step():
    neighbours = {}
    for x in range(-190, 190, 10):
        for y in range(-190, 190, 10):
            count = -cells[x, y]
            for h in [-10, 0, 10]:
                for v in [-10, 0, 10]:
                    count += cells[x+h, y+v]
            neighbours[x, y] = count
    for cell, count in neighbours.items():
        if cells[cell]:
            if count < 2 or count > 3:
                cells[cell] = False
            elif count == 3:
                cells[cell] = True
def draw():
    step()
    clear()
    for (x, y), alive in cells.items():
        color = 'green' if alive else 'black'
```

## » MORE DIMENSIONS

Of course, technology has come a long way since 1970, which Conway's *Game of Life* was first developed. Using specialist software and a high-end computer it's now possible to synthesise three-dimensional cellular automata, too. As well as checking neighbour cells on the x- and y-coordinates, we'd need to check the z-coordinates, so a slight adjustment to the rules would be necessary. Where Von Neumann considers eight local squares, a three-dimensional *Game of Life* needs to consider 26, based on a cuboid grid of 3x3x3. Three-dimensional Von Neumann methodology would mean every cell sharing a face with our centre cube is to be considered a neighbour. The updated rules are as follows:

- » A cell may survive if it has four neighbouring cells.
- » A cell is born when an empty location exists that has four neighbouring cells.
- » Alive cells with 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25 or 26 neighbours survive.
- » Empty cells with 13, 14, 17, 18 or 19 neighbours have a new cell born at that location.

There's also the option of having more than two states. In a traditional binary simulation a cell is either dead or alive, but in more complex simulations there can be five states, with '4' being a newly born cell and '0' being a completely empty cell. Cells fade down from 4 to 0. This offers better visuals, if nothing else.



An amoeba constructed in three dimensions, with Visions of Chaos.

```
color = 'green' if alive else 'black'
square(x, y, 10, color)
update()
ontimer(draw, 100)
setup(420, 420, 370, 0)
hideturtle()
tracer(False)
initialise()
draw()
done()
```

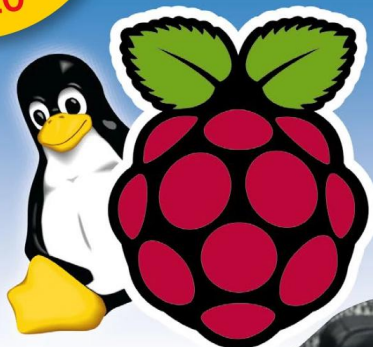
It's a game of chance. Sometimes you feel like you're discovering new life and the possibilities seem endless, whereas other times the game is over before you know it and the automata completely cease activity.

We haven't experimented with speeding up or slowing down our game, setting custom configurations or cell colouring, so there are plenty of areas for improvement. Do let us know how you get on, and if you construct any unique discoveries. **LXF**

» **KEEP LINUX FORMAT ALIVE!** Subscribe now at <http://bit.ly/LinuxFormat>



**LXF268**  
will be on  
sale Tuesday  
22 September  
2020



# A SMARTER HOME OFFICE

Build the perfect home-office with open  
source, a Raspberry Pi and Jonni!\*

## Linux on Windows

We take a dive into the Windows Subsystem for Linux 2.0, see what it can do and discover how far you can push it.

## Recreate Doctor Who!

Fire up your open source synth and recreate the analogue sounds of the long-running TV series for fun and games.

## Maths is fun!

Open source is blessed with an abundance of lab-ready mathematical applications – we test drive the best.

## Gaming on the Pi

Tired of building robots? Then why not install some classic games and use the Raspberry Pi for gaming fun!

Contents of future issues subject to change. \*Jonni not included: we need him to drink wine and know things.

# LINUX

The #1 open source mag

## FORMAT

Future Publishing Limited,  
Quay House, The Ambury, Bath, BA1 1UA  
Email [linuxformat@futurenet.com](mailto:linuxformat@futurenet.com)

### EDITORIAL

**Editor** Neil Mohr  
**Boat's afloat editor** Jonni Bidwell  
**Art editor** Efrain Hernandez-Mendoza  
**Operations editor** Cliff "italicise it all" Hope  
**Group editor in chief** Graham Barlow  
**Senior art editor** Jo Gulliver  
**Editorial contributors** Neil Bothwick, Mike McCallister, Nick Peers, Les Pounder, Calvin Robinson, Phil Savage, Shashank Sharma, Alexander Tolstoy

### Cover

Raspberry Pi is a trademark of the Raspberry Pi Foundation.  
Tux credit: Larry Ewing ([lewing@isc.tamu.edu](mailto:lewing@isc.tamu.edu)) and The GIMP.

### ADVERTISING

Media packs are available on request  
**Commercial sales director** Clare Dove  
[clare.dove@futurenet.com](mailto:clare.dove@futurenet.com)  
**Senior advertising manager** Lara Jaggon  
[lara.jaggon@futurenet.com](mailto:lara.jaggon@futurenet.com)  
**Head of commercial – Technology** Dave Randall  
[dave.randall@futurenet.com](mailto:dave.randall@futurenet.com)  
**Account director** Andrew Tilbury  
[andrew.tilbury@futurenet.com](mailto:andrew.tilbury@futurenet.com)

### INTERNATIONAL LICENSING

Linux Format is available for licensing. Contact the  
Licensing team to discuss partnership opportunities.  
**Head of Print Licensing** Rachel Shaw  
[licensing@futurenet.com](mailto:licensing@futurenet.com)

### SUBSCRIPTIONS & BACK ISSUES

**Web** [www.myfavouritemagazines.co.uk](http://www.myfavouritemagazines.co.uk)  
**Queries** [www.myfavouritemagazines.co.uk/en/contact](http://www.myfavouritemagazines.co.uk/en/contact)  
**UK** 0344 848 2852 **World** +44 (0) 344 848 2852

### CIRCULATION

**Head of newstrade** Tim Mathers

### PRODUCTION AND DISTRIBUTION

**Head of production UK & US** Mark Constance  
**Production project manager** Clare Scott  
**Advertising production manager** Joanne Crosby  
**Digital editions controller** Jason Hudson

### THE MANAGEMENT

**Chief content officer** Aaron Asadi  
**Editorial director** William Gannon  
**Brand director** Andy Clough  
**Head of art & design** Rodney Dive  
**Commercial finance director** Dan Jotcham  
**Printed by** Wyndeham Peterborough, Storey's Bar Road, Peterborough, Cambridgeshire, PE1 5YS  
**Distributed by** Marketforce, 5 Churchill Place, Canary Wharf, London, E14 5HU [www.marketforce.co.uk](http://www.marketforce.co.uk)  
**Tel:** 0203 787 9001

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.  
GNU/Linux is abbreviated to Linux throughout for brevity. Where applicable code  
printed in this magazine is licensed under the GNU GPL v2 or later. See [www.gnu.org/copyleft/gpl.html](http://www.gnu.org/copyleft/gpl.html). All copyrights and trademarks are recognised and respected.

**Disclaimer** All contents © 2020 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 cannot accept any  
responsibility for errors or inaccuracies in such information. You are advised to  
contact manufacturers and retailers directly with regard to the price of products/  
services referred to in this publication. Apps and websites mentioned in this  
publication are not under our control. We are not responsible for their contents or any  
other changes or updates to them. This magazine is fully independent and not  
affiliated in any way with the companies mentioned herein.

If you submit material to us, you warrant that you own the material and/or have  
the necessary rights/permissions to supply the material and you automatically grant  
Future and its licensees a licence to publish your submission in whole or in part in  
any/all issues and/or editions of publications, in any format published worldwide and  
on associated websites, social media channels and associated products. Any  
material you submit is sent at your own risk and although every care is taken, neither  
Future nor its employees, agents, subcontractors or licensees shall be liable for loss  
or damage. We assume all unsolicited material is for publication unless otherwise  
stated, and reserve the right to edit, amend, adapt all submissions. All contents in  
this magazine are used at your own risk. We accept no liability for any loss of data or  
damage to your systems, peripherals or software through the use of any guide.  
Notes: Neil has had his patio rebuilt!

We are committed to only using magazine paper which is derived  
from responsibly managed, certified forestry and chlorine-free  
manufacture. The paper in this magazine was sourced and  
produced from sustainable managed forests, conforming to strict  
environmental and socioeconomic standards. The manufacturing  
paper mill and printer hold full FSC and PEFC certification and  
accreditation.



Future is an award-winning international media group and  
leading digital business. We reach more than 57 million  
international consumers a month and create world-class  
content and advertising solutions for passionate consumers  
online, on tablet & smartphone and in print.

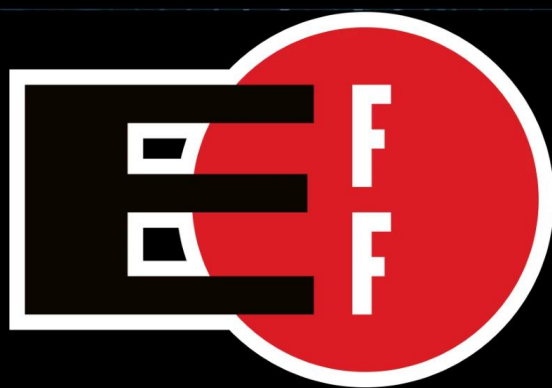


Future plc is a public  
company quoted on the  
London Stock Exchange  
(symbol: FUTR)  
[www.futureplc.com](http://www.futureplc.com)

Chief executive **Zillah Byng-Thorne**  
Non-executive chairman **Richard Huntingford**  
Chief financial officer **Rachel Addison**

Tel +44 (0)1225 442244





The Electronic Frontier Foundation is the leading nonprofit organization defending civil liberties in the digital world. Founded in 1990, EFF champions user privacy, free expression, and innovation through impact litigation, policy analysis, grassroots activism, and technology development. We work to ensure that rights and freedoms are enhanced and protected as our use of technology grows.

**EFF.ORG**

**ELECTRONIC FRONTIER FOUNDATION**

Protecting Rights and Promoting Freedom on the Electronic Frontier





**The  
Photography  
Show**

**The  
Video  
Show**

20 & 21 SEPTEMBER 2020 ONLINE EVENT

# WE'RE GOING VIRTUAL...

We're hosting a two-day virtual experience which will bring everyone passionate about photography and video together.

## BE PART OF SOMETHING DIFFERENT!

- **Check out the latest kit from, and chat to, your favourite brands - and some you may not yet know of**
- **Enjoy exclusive show offers - buy what you need (or want!) at fantastic prices**
- **Discover new skills and be inspired by our programme of tutorials, interviews and discussions**

...All from the comfort of your own home

**To register FREE, visit [photographyshow.com](https://photographyshow.com)**